VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2020 Bc. Andrzej Szymeczek

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

VYTVOŘENÍ SIMULAČNÍHO MODELU PŘÍSTUPOVÉ SÍTĚ
APPLICATION OF SIMULATION MODEL OF ACCESS NETWORK

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Andrzej Szymeczek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. David Grenar**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Andrzej Szymeczek ***ID:*** 186198
Ročník: 2 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Vytvoření simulačního modelu přístupové sítě

POKYNY PRO VYPRACOVÁNÍ:

Úkolem diplomové práce je provést rozbor vlastností simulačního nástroje GNS3, dále pak pomocí nástroje GNS3 navrhnout a realizovat model sítě a následně vytvořit simulaci pro testování zadané topologie sítě. Poté bude provedena instalace a konfigurace GNS3 s využitím aktivních prvků Mikrotik. Výstupem diplomové práce bude funkční simulační scénář zadané topologie sítě a nástroj pro generování provozů v simulačním prostředí.

DOPORUČENÁ LITERATURA:

[1] GNS3 | Graphic Network Simulator. [online]. Dostupné také z: https://www.gns3.com.

[2] WELSH, Chris. GNS3 Network Simulation Guide. 2013. Packt Publishing Limited, 2013. ISBN 9781782160809.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 1.6.2020

Vedoucí práce: Ing. David Grenar

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tématem diplomové práce je „Vytvoření simulačního modelu přístupové sítě“. Cílem práce bylo vytvořit část přístupové sítě, ve které se testovaly simulace pro TCP a UDP provoz. Teoretická část práce se věnuje obecným informacím o programu GNS3 a možnostech, které tento program umožňuje. Další část popisuje program Mikrotik RouterOS, který byl v topologii použit. V další části jsou popsány druhy směrování v IP sítích, a také typy doručování paketů. Praktická část se zabývá tvorbou zkušebních topologií a ověřováním výkonu simulačního nástroje. Dále se praktická část zabývá tvorbou síťového generátoru pro TCP a UDP provoz v síti. Generátor byl vytvořen pomocí Bash skriptu a kombinací příkazů dd a nc. V neposlední řadě práce popisuje výsledky simulací z pohledu přenosové rychlosti, propustnosti sítě, rozložení velikosti paketů a u TCP také obousměrné zpoždění – RTT.

KLÍČOVÁ SLOVA

GNS3, QEMU, Docker, NAT, RouterOS, TCP, UDP

ABSTRACT

The topic of the diploma thesis is " Application of simulation model of access network". The aim of the thesis was to create a part of the access network in which were tested simulations for TCP and UDP traffic. The theoretical part of the thesis deals with general information about the GNS3 program and the possibilities which this program allows. The next part of the thesis describes the Mikrotik RouterOS program, which was used in the topology. The next section describes the types of routing in IP networks, as well as the types of packet delivery. The practical part deals with the creation of test topologies and verification of the performance of the simulation tool. Furthermore, the practical part deals with the programming of a network generator for TCP and UDP traffic in the network. The generator was created by using a Bash script and a combination of the dd and nc commands. At the end, the thesis describes the results of simulations in terms of transmission speed, network throughput, packet size distribution and in the case of TCP also bidirectional delay – RTT.

KEYWORDS

GNS3, QEMU, Docker, NAT, RouterOS, TCP, UDP

SZYMECZEK, Andrzej. *Vytvoření simulačního modelu přístupové sítě*. Brno, Rok, 65 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. David Grenar

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Vytvoření simulačního modelu přístupové sítě“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Davidu Grenarovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále chci poděkovat všem blízkým, kteří v době tvorby celé práce stáli při mně a byli mi oporou.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tato diplomová práce se zabývá simulačním programem, který se jmenuje Graphical Network Simulator 3. V první kapitole jsou popsány základní informace, které přibližují program GNS3. Jsou zde uvedeny základní informace programu a jeho vlastnosti, se kterými je nutné se seznámit před zahájením práce s tímto simulátorem. V následující podkapitole jsou představeny emulátory, které jsou podporované programem GNS3. Jsou to emulátory: Dynamips, QEMU, VirtualBox, VMware, Docker, VPCS.

Další kapitola pojednává o typu směrovačů, které jsou v této práci používány, jedná se o směrovače společnosti Mikrotik s operačním systémem RouterOS. V této kapitole jsou popsány základní informace o Mikrotik směrovačích, a také základní funkce RouterOS, a to funkce směrování, DHCP a QoS.

Následující teoretická kapitola přibližuje směrování v IP sítích. Jsou zde popsány základní druhy směrování jako jsou: statické, výchozí a dynamické směrování. Poslední teoretická kapitola popisuje způsoby doručování paketů. Je zde popsán unicast, multicast, broadcast, a také anycast. U popisu multicastového vysílání jsou součástí protokoly pro podporu a správnou funkčnost.

Dále následuje praktická část práce. Nejdříve je specifikován hardware počítače, na kterém byly prováděny simulace. Za touto částí je popsáno srovnání výkonu při simulacích pod různými operačními systémy. Na základě tohoto srovnání se má vybrat operační systém, který je vhodnější pro simulace v programu GNS3. Následuje podkapitola o topologii sítě, kterou se vytvořilo. V této části jsou popsány všechny požadavky, které bylo třeba připravit pro tvorbu samotné topologie.

V další části práce je věnována síťovému generátoru pro dva základní přenosové protokoly TCP a UDP.

V poslední části práce jsou popsány výsledky simulací, kterých bylo dosaženo.

1 GRAPHICAL NETWORK SIMULATOR 3

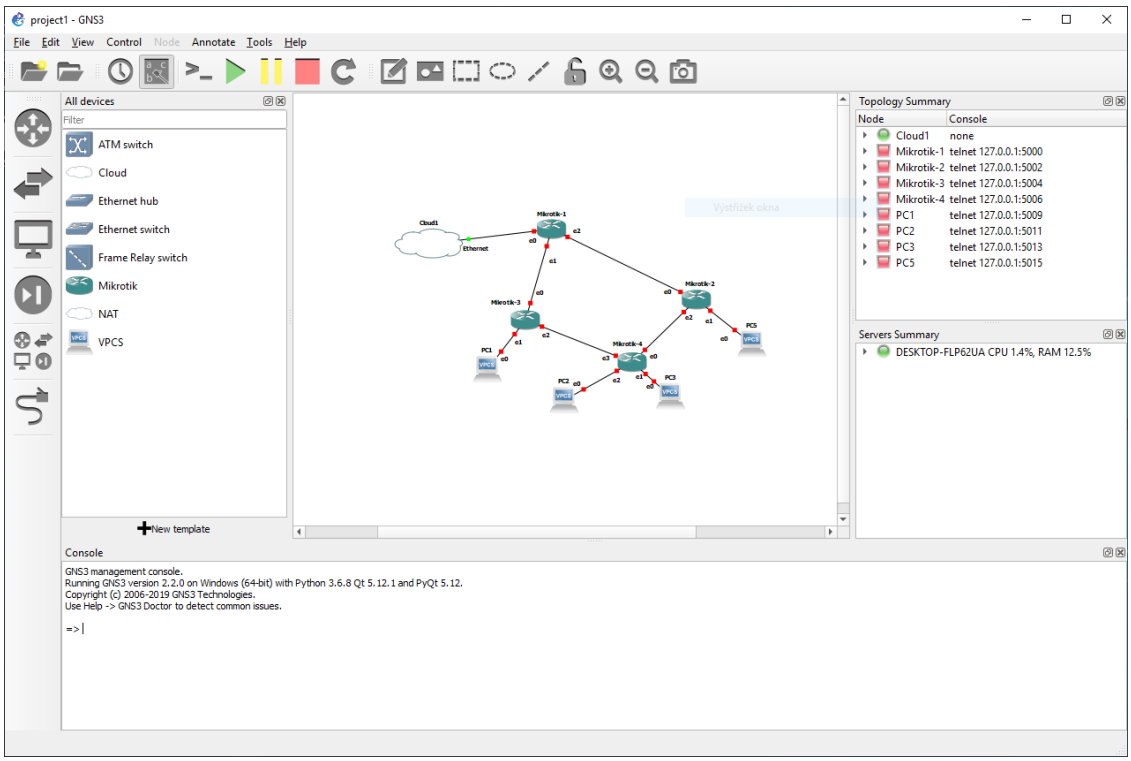
Simulátor GNS3 vznikl v roce 2007 a jeho autorem je Jeremy Grossman. Simulátor je program, který je opensource, a navíc zdarma. Za používání programu GNS3 není třeba nic platit jak v osobním, tak i komerčním prostředí. To je na rozdíl od konkurence značná výhoda. Opensource program umožňuje vždy zkontrolovat veškerý zdrojový kód, a také je možné se přidat do týmu vývojářů a přispět k dalšímu rozvoji programu. GNS3 je grafický síťový simulátor, který je schopen vytvořit, nakonfigurovat, testovat a opravovat chyby ve virtuálních i reálných sítích. Rozsah možností programu GNS3 je od malých topologií s několika zařízeními, až po velké topologie fungující na několika serverech či v cloudu. Také je zde možnost propojit existující fyzické sítě k vlastní topologii vytvořené v prostředí GUI GNS3. Program umožňuje virtualizaci reálných hardwarových zařízení jako jsou Dynamips, Cisco virtual switches, Cisco ASA, Brocade vRouters, Cumulus Linux switches, RouterOS a další. GNS3 se používá nejen pro studium, ale také pro testování sítí před jejich uvedením do výroby. Společnosti z celého světa testují sítě na GNS3 nejprve před nasazením. Možnosti a variace pro laboratorní topologie s GNS3 jsou téměř nekonečné. Další velkou výhodou tohoto simulačního nástroje je dán velkou aktivní komunitou lidí, kteří si navzájem pomáhají s tipy a triky, a také poskytují podporu v případě problémů. Členové komunity také přispívají laboratořemi, které mohou být využity ke studiu. Protože je GNS3 tak všestranný a výkonný, má tu nevýhodu, že je složité jej nastavit.

Program je podporován těmito operačními systémy:

- Windows 7 (64bit)
- Windows 8 (64bit)
- Windows 10 (64bit)
- Windows Server 2012 (64bit)
- Mac OS X Mavericks (verze 10.9 a novější)
- Linux

Program se skládá z dvou softwarových komponentů. První se jmenuje GNS3-all-in-one software (GUI) a druhý je GNS3 virtual machine (VM). GNS3-all-in-one software obsahuje grafické uživatelské rozhraní, které umožňuje rozsáhlé možnosti tvorby topologií nejrůznějších služeb dle potřeby obsluhy. Toto uživatelské rozhraní je zobrazeno na obrázku 1.1. Po vytvoření topologie a nastavení požadované konfigurace je možné simulovat chování topologie. Ke správnému fungování programu GNS3 musí existovat serverový proces, na kterém celá simulace a všechny její prvky běží. Existují tři možnosti, jak tento server může fungovat. První možnost je lokální GNS3 server, ten běží lokálně na stejném počítači jako program GNS3. Tato možnost je vhodná při tvorbě menších topologií. V tomto případě pro operační systém

Windows grafické rozhraní i lokální server poběží jako procesy v systému Windows. Další možnost fungování serveru je při použití lokálního GNS3 VM. Jedná se o virtuální stanici, kterou se dá spustit na lokálním počítači v některém z virtualizačních programů (například VMware, VirtualBox, Hyper-V), ta představuje server, na kterém poběží program. Poslední možnost vychází z té druhé. Jedná se o Remote GNS3 VM, tedy o virtuální stanici, na které běží virtuální stroj, který je geograficky na jiném místě než stanice, na které uživatel pracuje, ten může být zřízen pomocí VMware ESXi, nebo dokonce v cloudu [1, 2].



Obr. 1.1: Uživatelské rozhraní programu GNS3

Vývojáři programu GNS3 doporučují používat GNS3 VM pro většinu situaci, v případě, že program je nainstalován na operačním systému Windows nebo Mac OS. V případě používání programu GNS3 na operačním systému Linux toto doporučení přestává být důležité, a to z důvodu jiného přístupu operačního systému k procesům a hardwarovým prostředkům počítače [1, 2].

GNS3 neomezuje počet zařízení, která lze provozovat v topologii. Omezení představuje hardwarové prostředky, kterými disponuje uživatelská stanice. Další řešení, jako je Cisco VIRL, omezují počet zařízení v topologii na 20 zařízení Cisco (v závislosti na licenci). GNS3 to nedělá a lze provozovat stovky zařízení v topologii GNS3 (za předpokladu, že je dostupný dostatečný výkon hardwaru).

1.1 Emulátory podporované programem GNS3

GNS3 je simulátor, který se dá nazvat i emulátorem, a to z důvodu toho, že GNS3 emuluje hardwarové zařízení, které představuje skutečný obraz aktivního prvku. Můžete například zkopírovat reálny Cisco IOS z reálného fyzického routeru a spustit jej na virtuálním emulovaném routeru Cisco v GNS3. Oproti tomu simulace v GNS3 znamená, že simuluje vlastnosti a funkce zařízení, jako je přepínač. K tomuto se nepoužívají reálné operační systémy fyzického zařízení, ale simulované zařízení vyvinuté společností GNS3, jako je vestavěný přepínač druhé vrstvy. Emulátory GNS3 podporuje více emulátorů, které lze použít v projektech GNS3. To dává uživateli velkou flexibilitu při vytváření topologií. V této části jsou popsány jednotlivé emulátory [1, 2].

1.1.1 Dynamips

Dynamips je technologie využívaná v GNS3 již od počátku vývoje a emuluje směrovače Cisco a základní přepínání pomocí modulu Etherswitch. Emuluje starší hardware společnosti Cisco, například směrovače 3725 a používá skutečné obrazy Cisco IOS. Podporovaný obraz IOS se může zkopírovat z fyzického síťového zařízení a použít jej v GNS3. Obrazy IOS Cisco nejsou součástí GNS3, to znamená, že uživatel si obrazy musí obstarat sám. Společnost Cisco nepodporuje používání IOS obrazů na hardwaru jiných výrobců než Cisco. Dynamips emuluje platformy Cisco 1700, 2600, 2691, 3600, 3725, 3745 a 7200. Ačkoli původní vývoj Dynamips byl pozastaven od verze 0.2.8-RC2, vydané v říjnu 2007, vývoj pokračuje prostřednictvím týmu tvořícího program GNS3. Dynamips nyní pracuje na verzi 0.2.14-dev pro Windows, Linux a OS X a na verzi 0.2.8-RC2. Dynamips používá k emulaci velké množství paměti RAM a CPU. Pro obraz Cisco IOS, který vyžaduje 256 MB paměti RAM na skutečném 7200 routeru. Program přidělí toto množství virtuální instanci routeru. Dynamips také přiděluje určité množství paměti RAM pro ukládání překladů JIT do mezipaměti. V operačním systému Unix (ve výchozím nastavení) to představuje 64 MB paměti RAM a pro operační systém Windows je to 16 MB [3].

1.1.2 QEMU

Rychlý emulátor procesoru využívá dynamický překlad k dosažení dobré rychlosti emulace. Základní vlastnost emulátoru je to, že může běžet bez ovladače jádra hostitele, a přesto poskytuje přijatelný výkon. Využívá dynamický překlad do nativního kódu za přiměřené rychlosti, s podporou automatické úpravy kódu a přesných výjimek. Program je přenosný do několika operačních systémů (GNU / Linux, BSD,

Mac OS, Windows), a také architektur. Další vlastností je přesná softwarová emulace FPU [4].

QEMU obsahuje dva základní provozní režimy. První provozní režim je takzvaná plná emulace systému. V tomto režimu QEMU emuluje celý systém (například PC), včetně jednoho nebo několika procesorů, dále periferie, které lze použít ke spouštění různých operačních systémů počítače, nebo k ladění systémového kódu. V tomto provozním režimu může QEMU volitelně používat akcelerátor v jádře, jako je například KVM. Akcelerátory provádějí většinu kódu hosta nativně, zatímco nadále emulují zbytek stroje. Emulovat lze i různá hardwarová zařízení a v některých případech může hostující operační systém transparentně používat hostitelská zařízení, jako jsou například sériové, či paralelní porty, USB a další. Důležitou vlastností tohoto režimu je také podpora symetrického multiprocesování. Pro využití této podpory musí být součástí virtualizační akcelerátor, aby pro emulaci použil více než jeden hostitelský procesor. Druhý provozní režim se jmenuje emulace v uživatelském režimu. V tomto režimu může QEMU spouštět procesy kompilované pro jeden CPU na jiném CPU. Může být použit ke spuštění emulátoru rozhraní API systému Windows, nebo k usnadnění kompilace a ladění [5].

1.1.3 VMware / VirtualBox

Velice rozšířené a dobře využitelné programy pro virtualizaci zařízení. GNS3 podporuje přidávání svých virtualizovaných zařízení do topologií. Oba programy již v dnešní době obsahují podporu virtualizace VT-x pro procesory Intel a AMD-V pro procesory firmy AMD, což víceméně ponechává výběr mezi oběma programy pouze na osobních preferencích. Oba tyto programy nabízejí mnoho výhod pro použití v prostředí GNS3. Programy umožňují vytvářet komplexní topologie, které zahrnují servery a PC, které používají software společnosti Solarwinds a mnoho dalších dodavatelů, které lze integrovat přímo do topologií GNS3. Nutno podotknout, že GNS3 pouze integruje tyto dva virtualizační programy a nemá kontrolu nad konfigurací virtuálních strojů. Jakékoliv nastavení konfigurace virtuálních strojů je řízeno virtualizačním softwarem, nikoliv programem GNS3. To znamená, že projekty vytvořené v GNS3 jsou velice často obtížné k přenesení z jednoho počítače na druhý [6][7].

1.1.4 Docker

Jedná se o sadu platformy, která využívá virtualizaci na úrovni operačního systému k poskytování softwaru v balíčcích nazývaných kontejnery. Kontejnery jsou od sebe izolovány a sdružují svůj vlastní software, knihovny a konfigurační soubory. Ty mohou spolu komunikovat prostřednictvím definovaných kanálů. Všechny kontejnery

jsou provozovány jedním jádrem operačního systému, a jsou tedy odlehčeny oproti virtuálním strojům. Docker kontejner se stává kontejnerem, když běží na Docker Engine. Balíčkový software, který je k dispozici pro aplikace založené na Linuxu i Windows, bude vždy fungovat bez ohledu na infrastrukturu. Kontejnery izolují software od jeho prostředí a zajišťují, že pracuje jednotně i přes rozdíly například mezi vývojem a inscenováním.

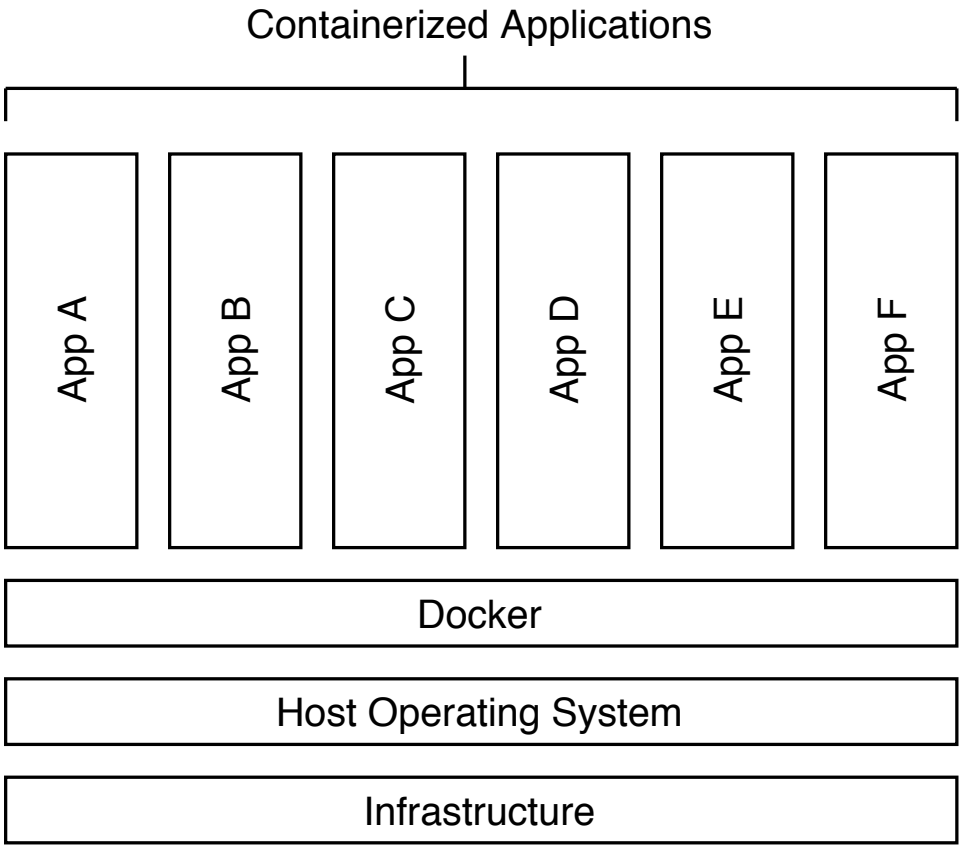
Technologie Docker kontejnerů byla zahájena v roce 2013 jako open source Docker Engine. Využíval existující výpočetní koncepty kolem kontejnerů a konkrétně ve světě Linuxu, primitivů známých jako cgroups a namespaces. Technologie společnosti Docker je jedinečná, protože se zaměřuje na požadavky vývojářů a provozovatelů systémů oddělit závislosti aplikací od infrastruktury. Úspěch ve světě Linuxu vedl k partnerství se společností Microsoft, které přineslo kontejnery Docker a jeho funkčnost na Windows Server někdy označované jako Docker Windows containers.

Docker kontejnery, které běží na Docker Engine se dělí na:

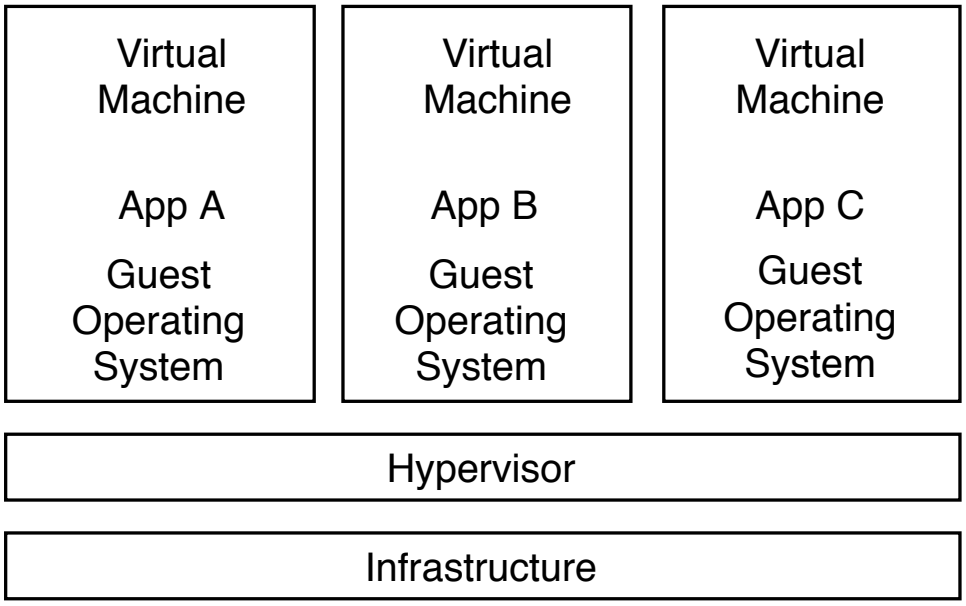
- Standard: Docker vytvořil průmyslový standard pro kontejnery, tak aby mohl být přenosný kdekoli
- Lehký: Kontejnery sdílejí jádro systému OS počítače, a proto nevyžadují OS pro aplikaci, zvyšují efektivitu serveru a snižují náklady na server a licencování
- Zabezpečené: Aplikace jsou v kontejnerech bezpečnější a Docker poskytuje nejsilnější výchozí izolační schopnosti v oboru

Porovnání kontejnerů a virtuálních strojů

Kontejnery a virtuální stroje mají podobné výhody izolace a alokace prostředků, ale fungují odlišně, protože kontejnery virtualizují operační systém namísto hardwaru. Kontejnery jsou přenosnější a efektivnější. Kontejnery jsou abstrakcí na aplikační vrstvě, která balí dohromady kód a programové souvislosti. Architektura kontejneru je zobrazena na obrázku 1.2. Na stejném počítači může být spuštěno více kontejnerů a sdílejí jádro OS s ostatními kontejnery, z nichž každý běží jako izolované procesy v uživatelském prostoru. Kontejnery zabírají méně místa než virtuální počítače (kontejnerů mají obvykle desítky MB), zvládnou více aplikací. Oproti tomu virtuální stroje (VM) jsou abstrakcí fyzického hardwaru, který mění jeden server na mnoho serverů. Architektura virtuálního stroje je zobrazena na obrázku 1.3. Hypervisor počítače umožňuje spuštění více VM na jednom počítači. Každý VM obsahuje úplnou kopii operačního systému, aplikace, potřebné binární soubory a knihovny, které zabírají i desítky GB. Zavádění virtuálních počítačů může být také pomalé.



Obr. 1.2: Architektura Docker kontejneru



Obr. 1.3: Architektura virtualizovaného systému

Docker je dobrá volba v případě, že chceme emulovat server nebo počítač poskytující konkrétní službu, jako je například server TFTP, poštovní server nebo webový server, a chceme toho dosáhnout bez použití většího využití paměti. Docker se stal populárním způsobem okamžitého vytvoření procesu nebo služby v porovnání s tradičními metodami zavádění celého virtuálního počítače za účelem poskytnutí jednotlivých služeb. V GNS3 je Docker používán k emulaci odlehčeného Linux PC s jednou službou. Často se Docker používá jako výkonná náhrada za VPCS [1].

1.1.5 VPCS – Virtual PC Simulator

VPCS je jednoduchý emulátor velmi základního PC. VPCS využívá velmi malé množství paměti RAM, a je proto dobrou volbou, pokud chceme emulovat počítač bez grafického uživatelského rozhraní, a pokud pro testování připojení v sítích GNS3 potřebujeme pouze jednoduché příkazy, jako je *ping*. Pro jakékoliv složitější servery nebo počítače je vhodné využít jiné emulátory jako jsou QEMU, VirtualBox, VMware [1].

2 MIKROTIK ROUTEROS

Mikrotik je lotyšská společnost, která se zabývá tvorbou síťových zařízení. Společnost Mikrotik vyrábí cenově dostupné síťové směrovače, a to jak kabelové, tak i bezdrátové. V portfoliu produktů této společnosti nechybí ani síťové přepínače a přístupové body. Dále pak firma tvoří operační systémy a pomocný software.

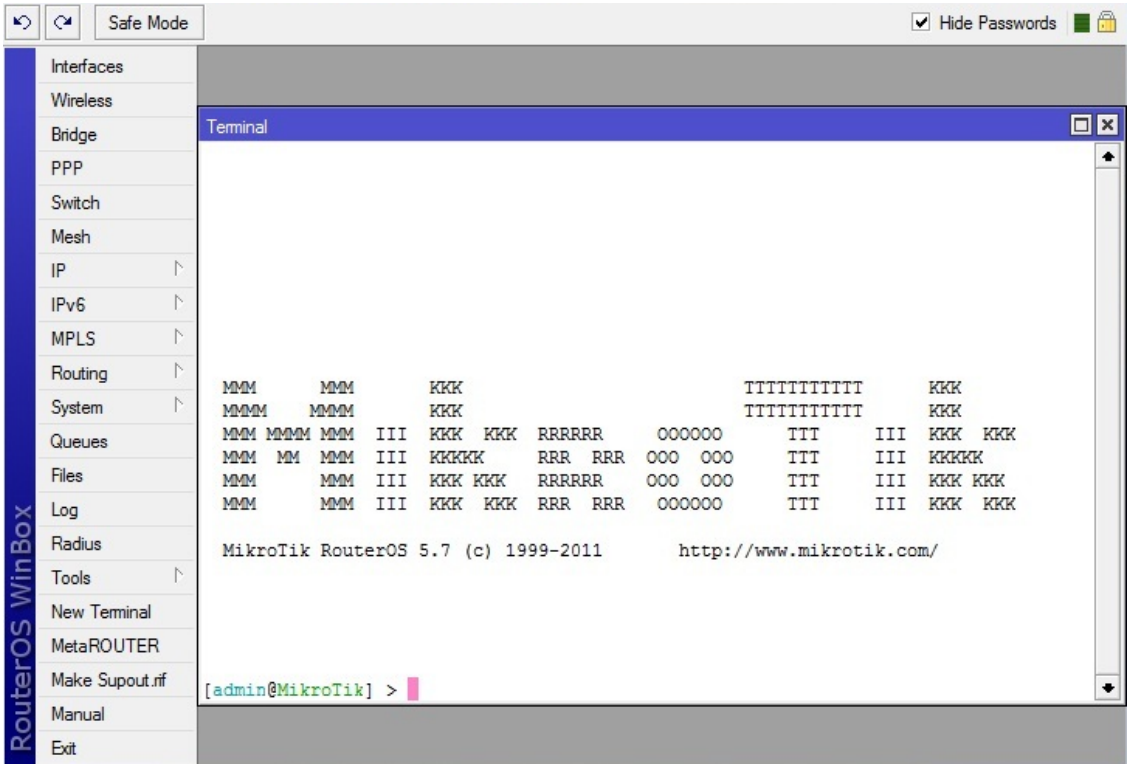
Veškerý hardware směrovačů Mikrotik patří do takzvané hardwarové platformy RouterBOARD, což jsou směrovače, které jsou přímo tvořeny pro operační systém RouterOS. Různé varianty RouterBOARD poskytují různé možnosti využití provozování, například bezdrátových přístupových bodů, spravování síťových přepínačů, dále pak zařízení s bránou firewall s podporou kvality služeb (QoS) [8].

Představitelem operačního systému vytvořeného společností Mikrotik je RouterOS. Tento operační systém je založený na základech platformy Linuxu a je přímo určen pro hardwarovou platformu RouterBOARD. RouterOS lze nainstalovat na PC a využít jej jako router s bránou firewall, serverem VPN a klientem či bezdrátovým přístupovým bodem. RouterOS je konfigurovatelný prostřednictvím příkazového řádku, sériového rozhraní, či telnetu. Další možností je konfigurace přes zabezpečenou komunikaci pomocí protokolu Secure Shell (SSH), či webového rozhraní (WebFig). Poslední možnost konfigurace je přes softwarovou aplikaci Winbox, která poskytuje grafické rozhraní a je tou nejběžnější variantou, která je využívána pro základní konfigurace operačního systému RouterOS. Grafické rozhraní Winbox aplikace je na obrázku 2.1.

Operační systém je volně přístupný, ale je zde možnost zakoupit různé úrovně licence, kdy každá úroveň licence přináší další funkce a výhody. Základní licence obsahuje funkce směrování, tvorbu a správu VLAN a další funkce. Omezení základní licence je v limitu maximálního množství tunelů (PPPoE, PPTP, L2TP, OVPN) a také je zde omezení, kdy HotSpot může mít maximálně jednoho aktivního uživatele. Vyšší licence tato omezení ruší až do nejvyšší licence, u které jsou všechny otevřeny, a navíc jsou všechny funkce neomezeny. K tomu je třeba přidat, že vyšší licence obsahují zaváděcí třiceti denní konfigurační podporu od výrobce [8][9].

2.1 Funkce RouterOS

RouterOS obsahuje velké množství funkcí, kterými lze spravovat sítě. V této podkapitole jsou v krátkosti popsány některé funkce, které jsou obsaženy v RouterOS.



Obr. 2.1: Winbox GUI

2.1.1 Funkce směrování

Mezi základní funkce patří funkce směrování. RouterOS podporuje statické směrování, Virtual Routing and Forwarding (VRF), Interface routing, při kterém není nutné zadávat adresu výstupní brány, ale pouze název rozhraní. Další významnou funkcí, kterou nabízí RouterOS je takzvané ECMP směrování. K implementaci některých nastavení jako je například vyvažování zátěže, kdy bude třeba použít více než jednu cestu k danému cíli. V jedné směrovací tabulce však není možné mít více než jednu aktivní trasu k cíli. Zde přichází směrování ECMP jejichž trasy obsahují více hodnot Nexthop brány. Všechny dostupné Nexthop brány jsou zkopírovány do Forwarding information base (FIB) a použity v předávacích paketech. Protokol OSPF může vytvářet trasy ECMP. Tyto trasy lze také vytvořit ručně. Důležitou součásti směrování v RouterOS jsou dynamické protokoly směrování pro IPv4. Protokoly, které se starají o tento typ směrování, jsou tato: RIP v1/v2, OSPFv2, BGP v4. Dále jsou zde dynamické protokoly pro IPv6 směrování: RIPng, OSPFv3, BGP. Poslední funkcí, která je zmíněna v této podkapitole je takzvané Bidirectional Forwarding Detection (BFD). Jedná se o protokol, který se stará o detekci obousměrného předávání. Jedná se o protokol s nízkými režijními náklady a s krátkodobým trváním určeným k detekci poruch v obousměrné cestě mezi dvěma předávacími stroji, včetně fyzických rozhraní, dílčích rozhraní, datových spojů s potenciálně velmi nízkou la-

tencí. Funguje nezávisle na médiích, datových protokolech a směrovacích protokolech. BFD je v podstatě hello protokol pro kontrolu dosažitelnosti obousměrného souseda. Poskytuje podporu detekce selhání linky po vteřině. BFD Control pakety jsou přenášeny v UDP paketech s cílovým portem 3784, BFD také používá port 4784 pro multihop cesty. Zdrojový port je v rozsahu 49152 až 65535. A pakety BFD Echo jsou zapouzdřeny v UDP paketu s cílovým portem 3785 [8] [9].

2.1.2 DHCP

RouterOS podporuje protokol DHCP pro dynamické přiřazování IP adres a dalších konfiguračních parametrů k jednotlivým síťovým zařízením, aby mohly mezi sebou komunikovat. Implementace MikroTik RouterOS zahrnuje serverové i klientské části a je v souladu s RFC 2131. DHCP v RouterOS umožňuje statické i dynamické časy DHCP výpůjček (leases). Router podporuje pro každý server samostatný server. Server DHCP podporuje základní funkce poskytující každému žádajícímu klientovi IP adresu / pronájem masky sítě, výchozí bránu, název domény, DNS-servery a WINS-servery. Pro správné fungování DHCP serveru musí být nakonfigurovány IP pools, což je rozsah adres, kterým DHCP server může distribuovat. Klient DHCP v RouterOS může být povolen na jakémkoli rozhraní. Klient přijme adresu, masku sítě, výchozí bránu a dvě adresy serveru DNS. Přijatá adresa IP bude přidána do rozhraní s příslušnou síťovou maskou. Výchozí brána bude přidána do směrovací tabulky jako dynamický záznam. Pokud bude klient DHCP deaktivován nebo neobnoví adresu, bude odstraněna dynamická výchozí cesta. Pokud je již před instalací klienta DHCP nainstalována výchozí trasa, bude trasa získaná klientem DHCP zobrazena jako neplatná [8] [9].

2.1.3 QoS

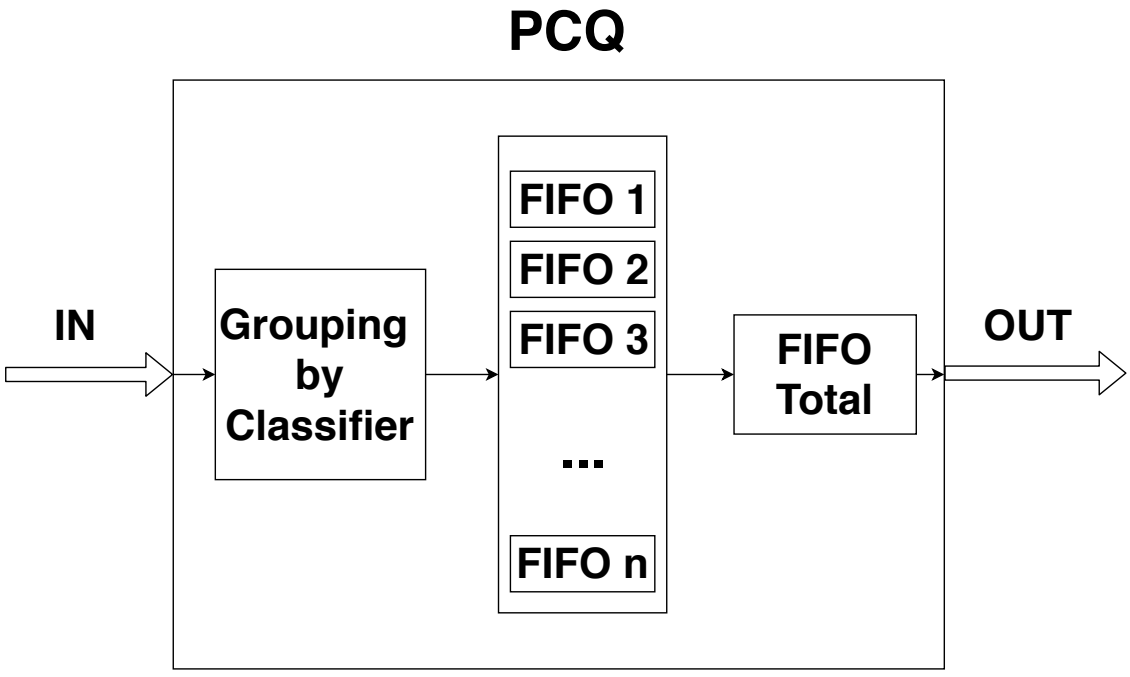
Kvalita služeb (QoS) označuje jakoukoli technologii, která řídí přenos dat, aby se snížila ztráta paketů, latence a jitter v síti. QoS řídí síťové zdroje stanovením priorit pro konkrétní typy dat v síti. Podnikové sítě musí poskytovat předvídatelné a měřitelné služby, protože aplikace jako například hlas, video a data citlivá na zpoždění procházejí sítí.

RouterOS nabízí několik variant, jak s QoS může pracovat. První varianta představuje takzvané jednoduché fronty. Jedná se o nejjednodušší způsob, jak omezit rychlost přenosu dat pro konkrétní adresy IP nebo podsítě. Další způsob, jak se RouterOS vypořádává se zajištěním kvality služeb je při použití Per Connection Queue (PCQ), což je způsob frontování, který lze použít k dynamickému vyrovnávání nebo tvarování provozu pro více uživatelů pomocí minimální správy. Funkčnost protokolu PCQ je zobrazena na obrázku 2.2. Scénáře PCQ lze rozdělit do tří hlavních

skupin: stejná šířka pásma pro více uživatelů, určitá šířka pásma a stejná distribuce mezi uživateli, neznámá šířka pásma a stejná distribuce mezi uživateli. Algoritmus PCQ je velmi jednoduchý. Nejprve používá vybrané klasifikátory k rozlišení jednoho dílčího proudu od jiného, poté aplikuje jednotlivé velikosti a omezení fronty FIFO na každý dílčí proud, a poté všechny skupiny spojuje dohromady a použije globální velikost a omezení front. Posledním představitelem QoS v RouterOS je Hierarchical Token Bucket (HTB). HTB umožňuje vytvořit hierarchickou strukturu fronty a určit vztahy mezi frontami, jako je "rodič-dítě"nebo "dítě-dítě". V RouterOS je nutné zadat nadřazenou možnost pro přiřazení fronty podřízené k jiné frontě. HTB je metoda řazení do fronty, která je užitečná pro zpracování různých druhů provozu v síti [8] [9].

Při vytváření HTB front musíme dodržovat tři základní kroky:

- Porovnat a označit provoz — roztřídění provozu pro další použití. Skládá se z jednoho nebo více odpovídajících parametrů pro výběr paketů pro konkrétní třídu.
- Vytvoření pravidel (zásad) pro označení provozu — vložení konkrétní třídy provozu do konkrétní fronty a definování následné akce, které se provádějí pro každou třídu.
- Připojení zásad pro konkrétní rozhraní



Obr. 2.2: Protokol PCQ

3 SMĚROVÁNÍ V IP SÍTÍCH

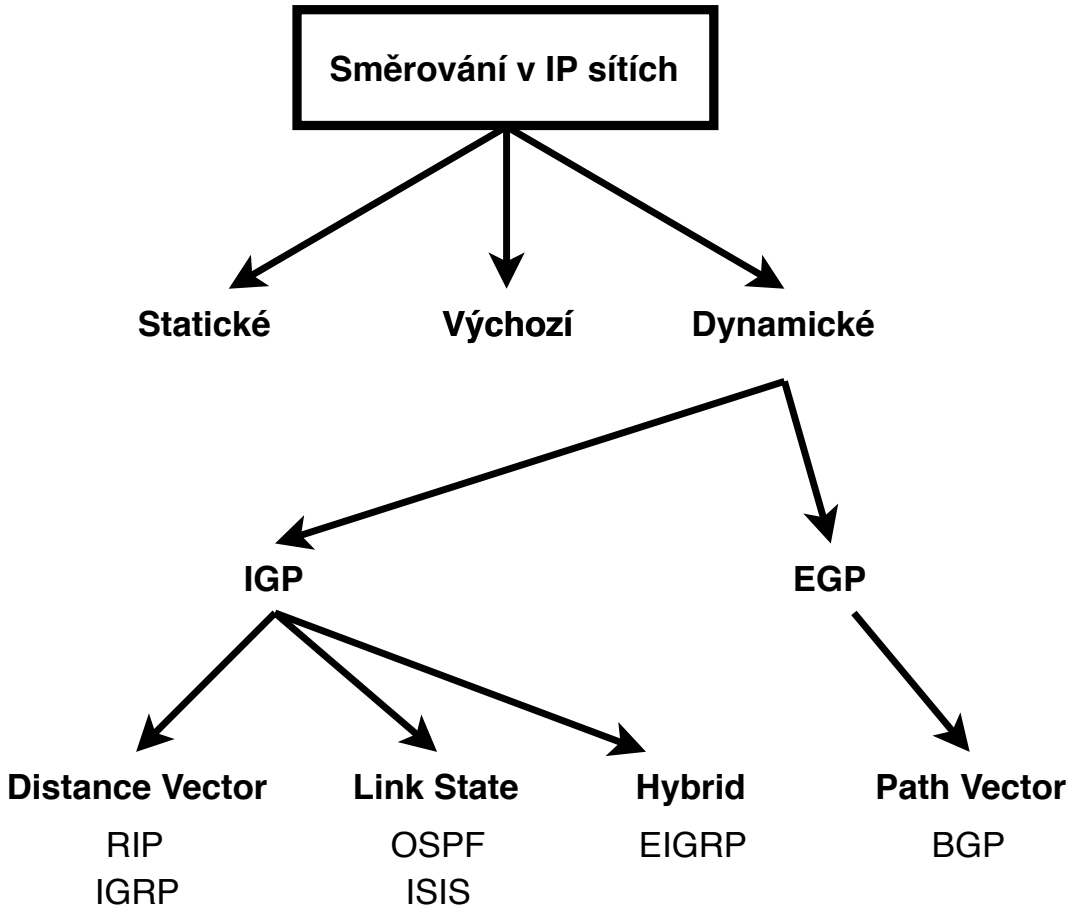
V této kapitole jsou popsány nejznámější a nejvíce používané směrovací protokoly. Routery mohou fungovat v rámci sítě za určitých pravidel. Jedno z pravidel je, aby každý aktivní prvek v sítí disponoval jedinečnou IP adresou v rámci sítě. Toto pravidlo a mnohé další je potřeba nastavit na všech aktivních prvcích v síti. Tato nastavování mohou zabírat veliké množství času, a proto vznikly směrovací protokoly, které představují možné řešení. Směrovací protokoly specifikují způsob, jakým budou routery mezi sebou komunikovat. Směrovače v síti určují, jakým směrem přicházející data budou dále posouvány od vysílače dat až ke koncové stanici. Síťová vrstva modelu OSI odpovídá za zajištění logického adresování, které směrovače používají k výběru nejlepší cesty pro směrování paketů. V této vrstvě se používají dva typy paketů:

- Datové pakety – Uživatelská data jsou těmito datovými pakety přenášena v síti. Směrované protokoly jsou protokoly, které podporují takový přenos dat. Příklady směrovaných protokolů jsou IPv4, IPv6.
- Pakety aktualizace trasy – Informace o sítích připojených ke všem směrovačům se aktualizují na sousední směrovače prostřednictvím paketů aktualizace trasy. Směrovací protokoly jsou ty, které jsou odpovědné za jejich odeslání. Příklady směrovacích protokolů jsou RIP (Routing Information Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol) a OSPF (Open Shortest Path First).

Routery používají směrovací protokoly k určení nejlepší cesty pro pakety k co nejefektivnějšímu směrování v síti. Směrovací protokoly jsou přiřazeny k rozhraní a určují způsob doručování paketů. Na obrázku 3.1 je znázorněno větvení, které popisuje typy směrování v rámci síťové vrstvy IP sítí [10] [11].

3.1 Statické směrování

První druh směrování je takzvané statické směrování. V tomto druhu směrování je potřeba ručně přiřadit všechny požadované trasy do směrovací tabulky. Výhodou tohoto způsobu směrování je, že nevzniká žádná režie pro procesor směrovače, což znamená, že lze použít levnější směrovač. Statické směrování navíc umožňuje přidat určitý druh zabezpečení, kdy pouze správce může povolit směrování do určitých sítí. Poslední zásadní výhodou je nulové využití šířky pásma mezi směrovači, co se týče přenosu směrovacích dat. Oproti tomu statické směrování má řadu nevýhod. První nevýhodou je větší množství směrovačů v síti. V tomto případě je velice pracné přidávat ručně každou trasu. Tento druh směrování typicky používají koncové stanice



Obr. 3.1: Směrovací protokoly

nebo routery v malých počítačových sítích (LAN), které se často nemění, což znamená jednorázové nastavení sítě a občasné úpravy, které nejsou nijak časově náročné [11].

3.2 Výchozí směrování (Default Routing)

Jedná se o metodu, kdy je router nakonfigurován tak, aby posílal všechny pakety směrem k danému směrovači (Nexthop). Nezáleží na tom, do které sítě paket patří, je předán routeru, který je nakonfigurován pro výchozí směrování. Tento typ směrování se nejčastěji používá u takzvaných Stub routerů, ten obsahuje pouze jednu cestu k dosažení všech ostatních sítí [11].

3.3 Dynamické směrování (Dynamic Routing)

Dynamické směrování umožňuje automatické přizpůsobení tras podle aktuálního stavu trasy ve směrovací tabulce. Dynamické směrování využívá protokoly k nalezení síťových cílů a jejich trasy, na kterých jsou připojeny. Představiteli tohoto typu směrování jsou protokoly OSPF a RIP. V případě, kdy dojde k přerušení spojení určité trasy, dojde k automatickému přizpůsobení pro dosažení cílové adresy. Mezi výhody tohoto typu směrování patří snadná konfigurace, účinnost při výběru a hledání nejlepších tras do cíle. Nevýhodami jsou pak větší spotřeba šířky pásma z důvodu větší komunikace mezi routery a menší bezpečnost oproti statickému směrování [11].

Dynamické směrování by mělo splňovat tyto vlastnosti:

- Všechny směrovače v rámci sítě by měly mít spuštěny stejný dynamický protokol, aby mohlo docházet k vzájemné komunikaci a výměně informací o trasách.
- Router, který zjistí nějakou změnu v síti, propaguje tuto informaci všem směrovačům.

3.3.1 Routing Information Protocol (RIP)

Jedná se o dynamický směrovací protokol, který patří do skupiny protokolu Interior Gateway Protocol (IGP) typu distance vector, který používá počet skoků jako směrovací metriku k nalezení nejlepší cesty mezi zdrojovou a cílovou sítí. Cesta s nejnižším počtem skoků k dosažení cílové sítě je považována za nejlepší cestu k dosažení cílové sítě, a proto je tato trasa umístěna ve směrovací tabulce. Směrovací protokol RIP má vnitřní mechanismy, které zabraňují vzniku smyček ve směrování, a to omezením maximálního počtu skoků na cestě ze zdroje do cíle. Maximální povolený počet skoků u protokolu RIP je 15 skoků. Je to protokol pro směrování vektorů vzdáleností a pracuje na aplikační vrstvě modelu OSI. RIP využívá číslo portu 520 a k přenosu využívá transportní protokol UDP. Protokol RIP pravidelně v daných intervalech aktualizuje změny v síti, tyto změny jsou vždy vysílány jako broadcast zprávy, což znamená, že jsou vysílány všesměrově do všech aktivních prvků sítě. Směrovače vždy důvěřují směrovacím informacím přijatým od sousedních směrovačů. Jednou z nevýhod protokolu RIP je vytváření a zbytečná spotřeba dostupné šířky pásma v důsledku periodicky vysílaných aktualizací, které se vyměňují mezi směrovači. Dále z důvodu vyměňování celých směrovacích tabulek přichází v úvahu problém se zabezpečením před možnými útoky [12].

3.3.2 Open Shortest Path First (OSPF)

Je směrovací protokol typu link– state, který k nalezení nejlepší cesty mezi zdrojovým a cílovým směrovačem používá SPF algoritmus. Tento algoritmus využívá koncept, který se spouští v případě aktualizace. Protokol je vyvinut společností Internet Engineering Task Force (IETF) jako jeden z protokolů Internal Gateway Protocol (IGP), tj. protokol, jehož cílem je pohyb paketu v rámci velkého autonomního systému nebo směrovací domény. OSPF používá vícesměrovou adresu 224.0.0.5 pro normální komunikaci a 224.0.0.6 pro aktualizaci na Designated Router (DR) nebo Backup Designated Router (BDR). Protokol OSPF vytváří síť sousedních směrovačů, pro které existují kritéria jejich vzniku. Oba směrovače by se měly nacházet ve stejné oblasti. Maska podsítě by měla být na obou směrovačích stejná. Časovače protokolu by měly být nastaveny na stejné hodnoty, navíc ověřování MD5 využité v protokolu OSPF musí vygenerovat shodu. Po splnění těchto kritérií je možné vytvořit dva sousedící směrovače. Protokol podporuje IPv4 i IPv6. OSPF využívá VLSM a sumarizaci trasy. Protokol také podporuje neomezený počet skoků [13].

OSPF využívá ke svému správnému fungování určité zprávy pro komunikaci mezi směrovači provozujícími protokol OSPF. Zde jsou krátce popsány nejdůležitější z nich:

- Hello message – Tato zpráva oznamuje směrovači objevení souseda nebo jeho znovu obnovení. Tyto zprávy se vyměňují ve výchozím stavu v intervalu 10 sekund. Tato zpráva také obsahuje tyto informace: Router ID, Router priority, DR a BDR IP adresy, autentizační data.
- Database Description (DBD) – Jedná se o OSPF trasy směrovače. Obsahuje topologii autonomního systému nebo oblasti.
- Link state request (LSR) – Požadavek na stav spojení. Směrovač příjme DBD zprávu, porovná ji s vlastním DBD a pokud přijatá DBD zpráva obsahuje více aktualizací než vlastní DBD, pak se zpráva LSR posílá sousedovi.
- Link state update (LSU) – Aktualizace stavu spojení. Když směrovač obdrží LSR zprávu, odpoví zprávou LSU obsahující požadované podrobnosti.
- Link state acknowledgement – Potvrzení stavu spojení. Odesílá se jako potvrzení LSU zprávy.
- Link state advertising (LSA) – Jedná se o datový paket obsahující informace o stavu spojení, který se sdílí pouze se směrovači, ke kterým bylo vytvořeno sousedství.

Součástí protokolu OSPF jsou již zmíněny časovače a zde jsou popsány:

- Hello timer – Interval, ve kterém OSPF směrovač odešle Hello message na požadované rozhraní. Ve výchozím stavu je tato hodnota rovná 10 sekund.
- Dead timer – Interval, ve kterém bude soused prohlášen za mrtvého, pokud

nebude schopen odeslat Hello message. Ve výchozím nastavení to je 40 sekund. Obvykle je tato hodnota čtyřnásobkem intervalu Hello timer [13].

3.3.3 Border Gateway Protocol (BGP)

Standardizovaný protokol ze skupiny protokolů EGP slouží k vyměňování směrovacích informací v rámci sítě mezi autonomními systémy. Jediným požadavkem je, aby každý AS měl alespoň jeden router, na kterém je funkční BGP a který je připojen k alespoň jednomu dalšímu BGP routeru jiného AS. Hlavní funkcí BGP je výměna informací o dosažitelnosti sítě s ostatními systémy BGP. Protokol BGP sestavuje graf autonomních systémů založený na informacích vyměňovaných mezi routery BGP. BGP je spojově orientovaný protokol, který používá TCP pro spolehlivé doručení.

Hlavními charakteristikami protokolu BGP jsou:

- Jedná se o směrovací protokol typu Path vector
- Používá se hlavně pro směrování provozu mezi autonomními systémy. To znamená, že hlavní použití je ve velkých internetových sítích.
- Protokol BGP používá časovače, aby se zabránilo neustálému inzerování rychle se měnící trasy napříč celým internetem.
- Vyměňovat směrovací informace mohou pouze ověřené směrovače, a tím se zaručí důvěryhodnost komunikujících stran.
- Mnoho vlastností fungování je potřeba nakonfigurovat ručně, včetně sousedů.

Zpočátku každý směrovač vytváří předpony, kterých může dosáhnout kromě svého vlastního čísla autonomního systému a tyto informace inzerují svým sousedům pomocí příznaku UPDATE zprávy. Když směrovač obdrží zprávu UPDATE, přidá nové trasy do své lokální směrovací tabulky na základě obsahu příchozí zprávy. Na druhou stranu se zařízení přidá do AS cesty před odesíláním dalších zpráv dalším zařízením. Když router zjistí porovnáváním, že jeho AS číslo již bylo zahrnuto do AS cesty, pak se směrovací informaci o trase odmítne, aby se zabránilo vytvoření směrovací smyčky. Protokol BGP dokáže detekovat směrovací smyčkou pouze mezi autonomními systémy, takže protokol nedokáže zabránit vytvoření smyček uvnitř AS. Ve zprávě UPDATE se kromě předpon posílá i informace o předvolbě cest, a také agregační informace.

Protokol BGP provádí tyto tři funkce na zařízeních:

- První funkce spočívá v počátečním získávání a ověřování jednotlivých zařízení. Zařízení navážou spojení TCP a provedou výměnu zpráv, která zaručuje, že se obě strany dohodly na komunikaci.

- Druhá funkce se zaměřuje hlavně na zasílání negativních nebo pozitivních informací o dosažitelnosti.
- Třetí funkce ověří, zda zařízení a síťové spojení mezi nimi fungují správně [14].

4 TYPY DORUČOVÁNÍ PAKETŮ

Rozlišujeme tři základní a jeden méně známý typ komunikace mezi systémy v IP sítích:

- Unicast
- Multicast
- Broadcast
- Anycast

4.1 Unicast

Unicast představuje typickou IP síť, která od zdroje vyšle paket, který je směrován jednomu přijímači. Většina datového provozu na internetu funguje na principu unicastu. Pokaždé, když uživatel navštíví web, existuje přímé spojení mezi klientem a serverem. Odesílání e-mailů také obvykle funguje prostřednictvím unicastu. Dalším příkladem použití je přímý přenos souborů. Například při streamování multimediálního obsahu se používají jiné metody, jako je multicast [15].

4.2 Multicast

Multicast je definován jako přenos, kdy existuje jediný zdroj dat odesílaný více příjemcům v síti. Multicast má svůj vlastní rozsah adresování. Pro multicastové vysílání se používá adresní rozsah třídy D, řízené a přiřazené úřadem IANA (Internet Assigned Numbers Authority). To znamená, že všechny vícesměrové vysílání IP jsou v rozsahu 224.0.0.0 až 239.255.255.255. Tento jedinečný rozsah adres IP se používá pouze pro cílovou adresu přenosu IP vícesměrového vysílání.

Multicastový datagram je dodáván členům cílové skupiny se stejnou spolehlivostí jako standardní unicastový IP datagram. To znamená, že multicastové datagramy nezaručují, že se dostanou ke všem členům skupiny nebo že dorazí ve stejném pořadí, v jakém byly předány. Jediným rozdílem mezi IP paketem jednosměrného vysílání a IP paketem vícesměrového vysílání je přítomnost skupinové adresy v poli cílové adresy IP záhlaví. Jednotliví hostitelé se mohou kdykoli připojit nebo opustit skupinu vícesměrového vysílání. Fyzická poloha ani počet členů ve skupině vícesměrového vysílání nepředstavují žádný problém. Hostitel může být členem více než jedné skupiny vícesměrového vysílání. Hostitel nemusí být členem skupiny, aby poslal pakety členům skupiny. Níže jsou popsány hlavní protokoly pro multicastový přenos.

4.2.1 Internet Group Management Protocol (IGMP)

Rozlišujeme tři verze tohoto protokolu. Následující podkapitola popisuje základní operace protokolu IGMP, společné pro všechny verze:

- Jeden směrovač periodicky vysílá zprávy IGMP Query na síťové propojení.
- Hostitelé reagují na zprávy Query zasláním zpráv IGMP Report s uvedením jejich členství ve skupině.
- Všechny směrovače přijímají zprávy Report a zaznamenávají členství hostitelů na síťovém spoji.
- Pokud směrovač neobdrží po určitou dobu zprávu Report pro určitou skupinu, předpokládá směrovač, že na síťovém propojení nejsou další členové skupiny.

Odesílání dotazů na členství ve skupině odesílá pouze jeden směrovač, ten odesílá zprávy IGMP Query na konkrétní síťový propoj. Tento router se nazývá Querier. Protokol IGMPv1 závisel na směrovacím protokolu vícesměrového vysílání, aby rozhodl, který router byl Querier. IGMPv2 představil Querierův volební proces, který funguje následovně. Ve výchozím stavu má směrovač roli IGMP Dotazatele (Querier), pokud dotazatel příjme zprávu IGMP Query ze směrovače na stejném rozhraní, a s nižší IP adresou, přestane být dotazatelem. Pokud směrovač přestál být dotazatelem, ale neobdrží IGMP Query zprávu v konfigurovaném intervalu, stane se opět dotazatelem. IGMPv1 a IGMPv2 používají techniku potlačení hlášení, aby se zabránilo „bouři“ odpovědí na zprávu IGMP Query. Když hostitel obdrží dotaz, spustí náhodný časovač pro každou skupinu, ve které je členem. Když se tento časovač objeví, hostitel odešle zprávu IGMP Report adresovanou této skupině. Všichni ostatní hostitelé, kteří jsou členy skupiny, také dostanou zprávu, kdy zruší svůj časovač pro skupinu. Tento mechanismus zajišťuje, že za většiny okolností je pro každou skupinu vícesměrového vysílání odesílána jedna zpráva IGMP jako odpověď na jeden dotaz. IGMPv3 odstranil tuto potřebu tím, že do jedné zprávy Report vložil více členství ve skupině, aby se snížil počet odeslaných paketů [15, 16].

4.2.2 Protocol Independent Multicast (PIM)

Jedná se o soubor protokolů vícesměrového směrování, z nichž každý je optimalizován pro jiné prostředí. Existují dva hlavní protokoly PIM, PIM Sparse Mode (řídký mód) a PIM Dense Mode (hustý mód). Třetí protokol PIM, obousměrný PIM, je méně používán. Obvykle bude v celé vícesměrové doméně používán buď řídký režim PIM, nebo hustý režim PIM. Mohou však být také použity společně v rámci jedné domény, řídký režim pro některé skupiny a hustý režim pro ostatní. Tato konfigurace ve smíšeném režimu se nazývá režim Sparse-Dense. Podobně lze použít obousměrný PIM samostatně, nebo může být použit ve spojení s jedním nebo oběma z PIM

řídkého režimu a PIM hustého režimu. Všechny PIM protokoly sdílejí běžný formát řídicí zprávy. Řídicí zprávy PIM jsou odesílány jako IP datagramy, a to buď vícesměrové vysílání do místní skupiny vícesměrového vysílání ALL PIM ROUTERS, nebo vícesměrové vysílání do konkrétního cíle [16].

PIM řídký režim (PIM-SM)

Směrovací protokol vícesměrového vysílání navržený za předpokladu, že příjemci pro každou konkrétní skupinu vícesměrového vysílání budou rozptýleni po celé síti. Jinými slovy se předpokládá, že většina podsítí v síti nebude chtít žádný daný paket vícesměrového vysílání. Aby bylo možné přijímat data vícesměrového vysílání, musí směrovače výslovně sdělit svým sousedům jejich zájem o konkrétní skupiny a zdroje. Směrovače používají zprávy PIM Join a Prune k připojení a opuštění stromů vícesměrového vysílání.

PIM-SM ve výchozím nastavení používá sdílené stromy, což jsou distribuční stromy vícesměrového vysílání zakořeněné v některém vybraném uzlu (v PIM se tento směrovač nazývá Rendezvous Point nebo RP) a používají je všechny zdroje odesílané do skupiny vícesměrového vysílání. Chceme-li odeslat do RP, musí zdroje data zapouzdřit do řídících zpráv PIM a odeslat je do RP. To se provádí pomocí určeného směrovače zdroje (DR), což je směrovač v místní síti zdroje. Jeden DR je zvolen ze všech PIM routerů v síti, takže nejsou zasílány zbytečné řídicí zprávy. Jedním z důležitých požadavků režimu PIM Sparse Mode a obousměrného PIM je schopnost zjistit adresu RP pro skupinu vícesměrového vysílání pomocí sdíleného stromu. Používají se různé mechanismy objevování RP, včetně statické konfigurace, Bootstrap router, Auto-RP, Anycast RP a Embedded RP.

PIM-SM také podporuje použití stromů založených na zdrojích, ve kterých je pro každý zdroj odesílající data do skupiny vícesměrového vysílání vytvořen samostatný distribuční strom vícesměrového vysílání. Každý strom je zakořeněn ve směrovači sousedícím se zdrojem a zdroje odesílají data přímo do kořenového stromu. Stromy založené na zdrojích umožňují použití zdroje specifického vícesměrového vysílání (SSM), které umožňuje hostitelům určit zdroj, ze kterého chtějí přijímat data, a také skupinu vícesměrového vysílání, ke které se chtějí připojit. SSM hostitel identifikuje vícesměrový datový tok s párem zdrojové a skupinové adresy (S, G), spíše než podle samotné skupinové adresy (*, G).

PIM hustý režim (PIM-DM)

Směrovací protokol vícesměrového vysílání navržený s opačným předpokladem než PIM-SM, konkrétně to, že přijímače pro jakoukoli skupinu vícesměrového vysílání jsou distribuovány hustě v celé síti. To znamená, že se předpokládá, že většina (nebo

alespoň mnoho) podsítí v síti bude chtít jakýkoli daný multicastový přenos. Data vícesměrového vysílání se zpočátku odesílají všem hostitelům v síti. Směrovače, které nechtějí být součástí stromu, odešlou zprávy PIM Prune a odstraní se ze stromu.

Když zdroj poprvé začne odesílat data, každý směrovač v zdrojové LAN data přijme a předá je všem svým sousedům PIM a všem linkám přímo připojenými k přijímači pro data. Každý směrovač, který přijímá přeposlaný paket, jej také přeposílá, ale pouze po kontrole, že paket dorazil na své předcházející rozhraní. Pokud ne, paket je vynechán. Tento mechanismus zabraňuje předávání smyček. Tímto způsobem jsou data zaplavena do všech částí sítě.

Některé směrovače nebudou vyžadovat data ani pro přímo připojené přijímače, ani pro další PIM sousedy. Tyto směrovače reagují na příjem dat zasláním zprávy PIM Prune upstream, která provede stav Prune v routeru, což způsobí, že přestane předávat data svému sousedovi.

PIM-DM používá pouze stromy založené na zdrojích. Výsledkem je, že nepoužívá RP, což usnadňuje implementaci a nasazení než PIM-SM. Je to efektivní protokol, když se většina přijímačů zajímá o data vícesměrového vysílání, ale nedochází k dobrému škálování napříč většími doménami, ve kterých většina přijímačů nechce být součástí vícesměrových skupin.

PIM Obousměrný režim (BIDIR-PIM)

Obousměrný PIM je třetí protokol PIM založený na PIM-SM. Hlavní způsob, jakým se BIDIR-PIM liší od PIM-SM, je metoda používaná k odesílání dat ze zdroje do RP. Zatímco v PIM-SM jsou data odesílána pomocí zapouzdření nebo stromu založeného na zdrojích, v BIDIR-PIM proudí data do RP podél sdíleného stromu, který je obousměrný. Datové toky proudí v obou směrech podél dané větve.

Hlavní rozdíly mezi BIDIR-PIM a PIM-SM jsou tyto:

- Neexistují žádné zdrojové stromy. Proto neexistuje možnost pro směrovače přepnout ze sdíleného stromu na zdrojový strom, a také zdrojově specifický multicast není podporován.
- Aby nedocházelo k předávání smyček, je pro každý RP zvolen jeden směrovač na každém spojení jako určený směrovač (Designated Forwarder – DF). To se provádí v době RP zjišťování pomocí DF volební zprávy (DF election message).
- Neexistuje žádný koncept určeného směrovače (Designated router).
- Nepoužívá se zapouzdření.
- Pravidla pro předávání jsou mnohem jednodušší než v PIM-SM a v řídicí rovině nejsou vůbec žádné řídící data.

Hlavní výhodou BIDIR-PIM je to, že se škáluje velmi dobře, když pro každou skupinu existuje mnoho zdrojů. Nedostatek stromů založených na zdrojích však zna-

mená, že provoz je nucen zůstat na možná neúčinném sdíleném stromu [15, 16, 17].

4.3 Broadcast

Broadcast, neboli všesměrové vysílání znamená, že síť doručí jednu kopii paketu do každého cíle. U sběrnicových technologií, jako je Ethernet, může být přenos vysílání uskutečněn pomocí jediného paketového přenosu. V sítích sestávajících z přepínačů s připojením point-to-point musí software implementovat vysílání předáváním kopií paketu přes jednotlivá připojení, dokud všechny přepínače neobdrží kopii.

Za normálních okolností, když počítač v síti obdrží paket, nejprve se pokusí porovnat MAC adresu paketu s jejich vlastní, a pokud je toto porovnání úspěšné, paket zpracují a předají je vyšší OSI vrstvě, pokud adresa MAC není uzavřena, paket je vyřazen a není zpracován. Když však počítač obdrží MAC adresu FF:FF:FF:FF:FF:FF, budou tento paket zpracovávat, protože jej rozpoznají jako všesměrové vysílání. Adresa IP všesměrového vysílání zajišťuje, že bez ohledu na to, jakou adresu IP přijímající počítač (počítače) mají, data neodmítnou, ale zpracují je.

Všesměrové vysílání se používá v různých síťových protokolech. Příklad tvoří například protokol ARP – Address Resolution Protocol. Tento protokol se používá ke zjištění, která MAC adresa má na ni vázanou konkrétní IP adresu. Další příklad využití představuje protokol DHCP. V tomto případě se DHCP server ptá na výpůjčku konkrétní IP adresy napříč sítí. V normální situaci by protokol DHCP musel obsílat všechny účastníky sítě přes unicast. V tomto případě je mnohem jednodušší, že pomocí broadcastu je vyslán dotaz automaticky na všechny stanice sítě. Všesměrové vysílání rozlišuje dva základní typy: omezené všesměrové vysílání a řízené všesměrové vysílání.

4.3.1 Omezené všesměrové vysílání

Při omezeném všesměrovém vysílání je jako cíl zadána adresa IP. Tato adresa IP je vždy 255.255.255.255. Technicky by toto všesměrové vysílání mělo být zasláno na všechny existující adresy IP. Ve skutečnosti však slouží jako adresa pro vysílání v síti. Tento cíl je vždy ve své vlastní síti, a proto může být implementován do Ethernetového všesměrového vysílání. Směrovač takový paket nepředává dál.

4.3.2 Řízené všesměrové vysílání

Při řízeném všesměrovém vysílání jsou všichni příjemci vždy osloveni v cílové síti. Kombinace čísla cílové sítě a nastavení všech bitů hostitele na 1 vytvoří v tomto

případě adresu vysílání. Pokud cíl není umístěn ve vlastní (pod) síti, směrovač předá datový paket.

Hostitelské bity jsou součástí IP adresy identifikující specifického hostitele v podsíti. Maska podsítě určuje, jaká část adresy se použije pro síťové bity, a které pro hostitelské bity. Například adresa IPv4 192.168.0.64/26 má 6bitovou hostitelskou část, protože 26 z 32 bitů je vyhrazeno pro síťovou část [16].

4.4 Anycast

Anycast je síťová technika, která umožňuje více strojům sdílet stejnou IP adresu. Na základě umístění požadavku uživatele jej směrovače odešlou do zařízení v síti, která je nejblíže. To je výhodné, protože to mimo jiné snižuje latenci a zvyšuje redundanci. Pokud by se určité datové centrum mělo přepnout do režimu offline, IP adresa zařazena do anycastu by vybrala nejlepší cestu pro uživatele a automaticky je přesměroval do nejbližšího datového centra. Následující text nastiňuje některé výhody a nevýhody spojené s konfigurací anycastu.

Výhody:

- Rychlost. Provoz směřující do libovolného uzlu bude směrován do nejbližšího uzlu, čímž se sníží latence mezi klientem a samotným uzlem. Tím je zajištěno, že rychlosti budou optimalizovány bez ohledu na to, odkud klient požaduje informace.
- Redundance. Anycast zlepšuje redundanci umístěním více serverů po celém světě pomocí stejné IP. To umožňuje přesměrování provozu na další nejbližší server v případě, že jeden server selže nebo přejde do režimu offline.
- Zmírnění DDoS. Útoky DDoS jsou způsobeny botnety, které mohou generovat tolik provozu, že přemohou typický unicastový stroj. Výhodou konfigurace anycast v této situaci je, že každý server je schopen „absorbovat“ část útoku, což má za následek menší zatížení serveru.
- Vyrovnávání zatížení. Vyrovnávání zátěže lze využít v případě, že existuje více uzlů ve stejné geografické vzdálenosti od žádosti. Tím se odstraní některé požadavky na zdroje z jednotného uzlu a rozptýlí se na více uzlů.

Hlavním problémem anycastového vysílání je, že redundance, kterou poskytuje, není vždy bezchybná; je možné, že server částečně selže, a přitom se stále zobrazuje jako „dostupný“ v síti, což znamená, že síť není dostupná těm, kteří jsou geograficky nejblíže problému. Správná funkce Anycast „prezenčního signálu“, která monitoruje servery a odstraňuje je ze sítě při jakémkoli náznaku problému, může tento problém ve většině případů zmírnit. Kromě toho je požadována investice větší než prosté použití jednosměrného vysílání k odesílání a přijímání dat z jednoho uzlu. Nevýhodou Anycastu je tím pádem náročnost jeho implementace [16].

5 SIMULAČNÍ SCÉNÁŘ TOPOLOGIE SÍTĚ

V této kapitole bude popsána praktická část práce. Kapitola je rozdělena do podkapitol, kde nejdříve je popsán hardware počítače, na kterém byly simulace prováděny. V další části je provedeno srovnání, při kterém se testovalo propojení hardwaru s operačními systémy při provádění simulací v prostředí GNS3. Toto srovnání testovalo chování a výkonnost simulací v programu GNS3 pod různými operačními systémy. Jednalo se o srovnání mezi OS Windows 10 a OS Ubuntu. V další podkapitole je popsána vytvořená funkční topologie sítě, která představuje model přístupové sítě, dále jsou pak popsány konfigurace, které se prováděly na jednotlivých směrovačích.

5.1 Hardware počítače pro simulace

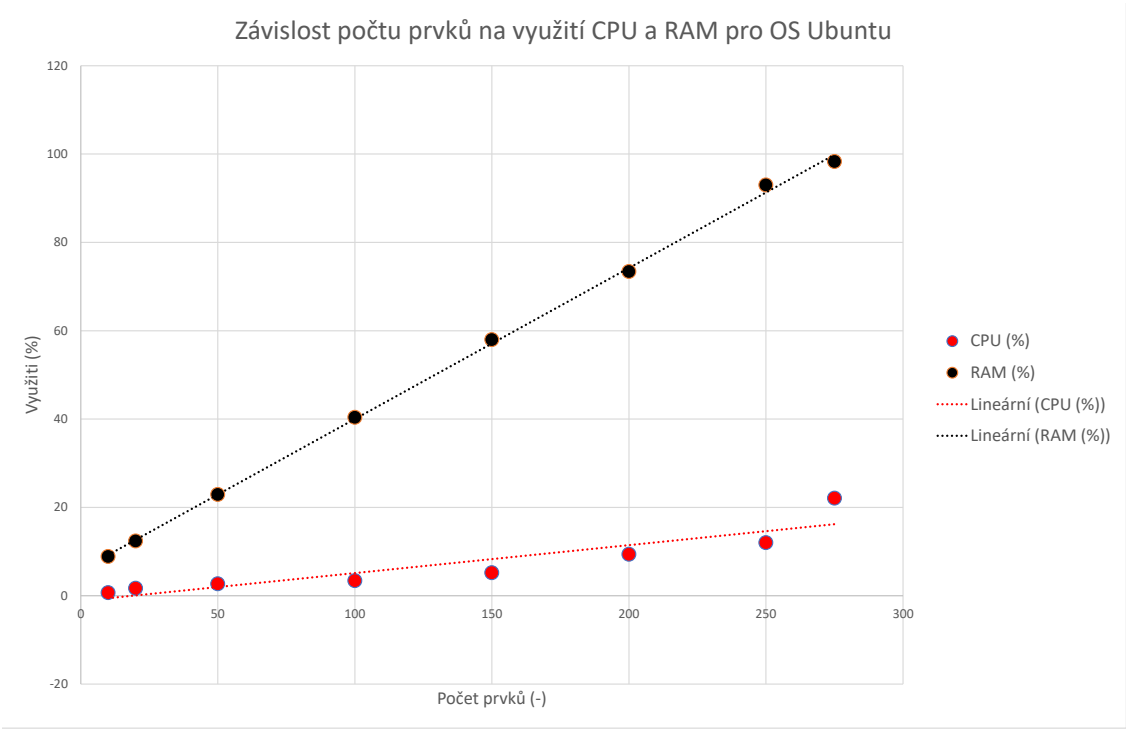
Pro účely této práce byl poskládán počítač, který byl přizpůsoben pro účely, které jsou vhodné při provádění simulací. Jádro počítače představuje procesor firmy AMD, konkrétně procesor AMD RYZEN 7 2700X, který obsahuje 8 fyzických jader, které pracují na základním taktu 3,7 GHz, ale je zde i možnost přetaktování na maximální frekvenci 4,3 GHz. Díky funkci Multi–threading procesor obsahuje 16 vláken. Tento procesor ve své cenové kategorii svými vlastnostmi překoná i procesory od společnosti Intel, a to z důvodu většího počtu jader a vláken. Procesory řady RYZEN 7 přináší vcelku novou architekturu procesoru, která je konkurence schopná procesorům společnosti Intel. Procesor také obsahuje podporu virtualizace AMD-V, která přináší výkon navíc, a tím má výhody pro účely simulací a provozování virtuálních strojů. Tato vlastnost procesoru byla rozhodující při výběru procesoru právě firmy AMD. Dále je součástí počítače paměti RAM, tu představují dva kusy po 16 GB operační paměti, tyto paměti pracují na frekvenci 3200 MHz a jsou to paměti typu DDR4. Tyto paměti dosahují propustnosti až 25600 MB/s. Data jsou v počítači ukládány na SSD disk, jehož kapacita je 500 GB. Poslední komponentou je základní deska, která obsahuje vhodnou patici pro procesor, v tomto případě se jedná o patici AM4. Dále základní deska obsahuje podporu zvolených operačních paměti. Všechny komponenty byly osazeny do skříně počítače a zprovozněny. Následně proběhlo otestování hardwaru. Na počítači jsou nainstalovány dva operační systémy (Windows 10 a Ubuntu).

GNS3 v prostředí OS Ubuntu		
Počet aktivních prvků	**Využití CPU (%)**	**Využití RAM (%)**
10	0,7	8,9
20	1,7	12,4
50	2,7	22,9
100	3,4	40,4
150	5,2	58
200	9,4	73,4
250	12	93
275	22,1	98,3

Tab. 5.1: Využití procesoru a paměti RAM v operačním systému Ubuntu

5.2 Srovnání výkonu při simulacích GNS3 v OS Windows 10 a OS Ubuntu

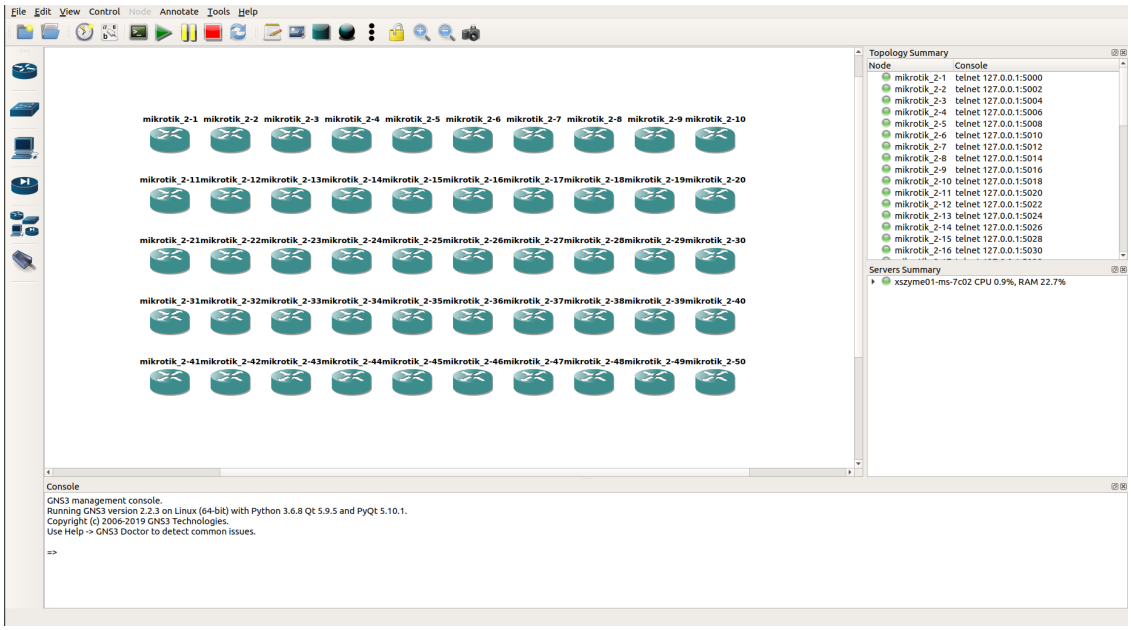
Tato podkapitola popisuje výsledky srovnání výkonu síťového simulátoru GNS3 v závislosti na operačním systému a na základě tohoto srovnání výkonu se dále určilo, na kterém operačním systému se bude provádět další simulace. Oba operační systémy se liší v mnoha ohledech, a také je třeba zohlednit způsob, kterým je navržen program GNS3. Proto se provedlo testovací měření, kdy se vytvořilo projekty. Tyto projekty obsahovaly různý počet aktivních emulovaných směrovačů firmy Mikrotik. Množství aktivních prvků se přidávalo do doby, než bylo vytížení procesu či paměti RAM v hraničních hodnotách svého fungování. Tyto směrovače byly emulovány virtualizačním softwarem Qemu. Každý jeden spuštěný směrovač Mikrotik přestavuje jeden virtuální stroj Qemu, pro který si vymezuje určité množství paměti RAM, a také daný počet jader procesoru, které může stroj využívat. V případě směrovačů Mikrotik bylo přiřazeno 256 MB paměti RAM a jedno jádro procesoru. Jedná se o maximální hodnotu prostředků, kterou může daný stroj využít. Na těchto směrovačích se neprováděla žádná konfigurace zařízení, cílem tohoto testu bylo zjištění, který operační systém lépe pracuje s hardwarovými prostředky, a to hlavně s výpočtovým výkonem procesoru a s pamětí RAM. Výsledky, které jsou níže popsány zahrnují celkové využití procesoru a paměti RAM včetně prostředků, které využívá samotný operační systém ke svému chodu.



Obr. 5.1: Závislost počtu aktivních prvků na využití CPU (%) a RAM (%) pro OS Ubuntu

Nejdříve proběhlo otestování možností rozsahu topologií v operačním systému Ubuntu. Na obrázku 5.2 je topologie obsahující 50 směrovačů Mikrotik spuštěných najednou. V tabulce 5.1 a na obrázku 5.1 je zobrazena závislost, jak se mění využití CPU a paměti RAM v závislosti na množství spuštěných směrovačů Mikrotik. Operační systém Ubuntu dovoluje vytvořit v prostředí GNS3 až 275 aktivních prvků. Při tolika prvcích dochází k 98,3 % využití paměti RAM a procesor byl vytížen na 22,1 %. Nutno podotknout, že při spuštění všech prvků daného scénáře došlo k 100 % využití procesoru, kdy se spouštěly procesy virtualizace směrovačů, dále pak dochází ke kopírování dat do operační paměti. Po krátké době došlo k ustálení využití procesoru a paměti RAM, přičemž se využití procesoru razantně snížilo. Pro reálnou simulaci s konfiguracemi směrovačů, daty proudícími přes síť, je 275 směrovačů za hranou dobré funkčnosti celé simulace, a to z důvodu skoro nulové rezervy hardwarových prostředků počítače, jelikož další spuštěné aplikace jako je generátor dat, konfigurace zabírají další prostředky, na které už ovšem daný hardware nemá kapacitu, a mohlo by docházet ke zkreslení výsledků, které jsou zajisté nežádoucí. Pro dobrý chod simulace je vhodné zachovat určitou rezervu v hardwarových prostředcích v řádách několika gigabajtů paměti RAM. V této hardwarové konfiguraci a dle výsledků je vhodné v topologiích použít maximálně 200 směrovačů Mikrotik. Pro 200 směrovačů Mikrotik docházelo na OS Ubuntu k 9,4 % využití procesoru a

k 73,4 % využití paměti RAM, což činí využití RAM 23,2 GB z 31,6 GB

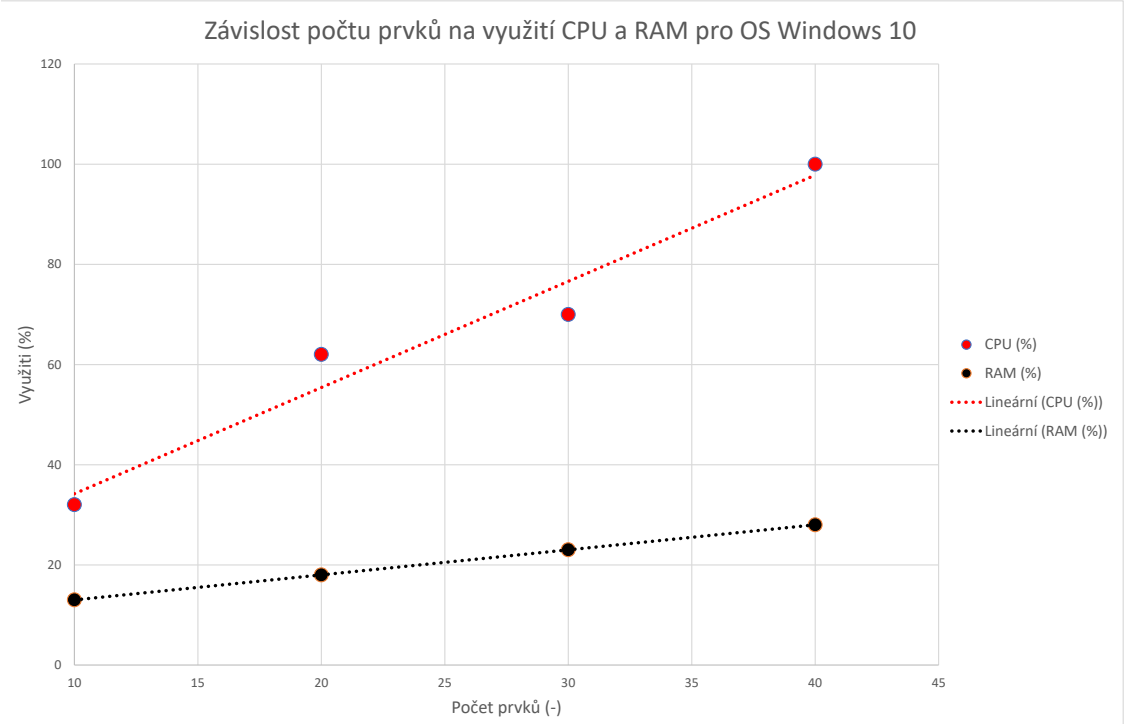


Obr. 5.2: Ubuntu - 50 směrovačů Mikrotik

Další tabulka 5.2 a grafická závislost 5.3 zobrazuje závislost na platformě operačního systému Windows 10. V prostředí operačního systému Windows 10 je vidět razantní rozdíl oproti simulaci na operačním systému Ubuntu. Operační systém Windows 10 hardwarově umožnil zprovoznit maximálně 40 aktivních prvků. Při 40 prvcích byl procesor vytížen na 100 % a paměť RAM byla vytížena na 28 %. Využití paměti RAM je ve výsledku podobná jako u testu na OS Ubuntu, rozdíl ve využití paměti RAM je v množství využití paměti samotným operačním systémem, kdy operační systém Windows 10 vyžaduje větší množství operační paměti ke svému chodu. Podobnost ovšem nelze přisoudit využití procesoru, kdy je zde vidět razantní rozdíl. V OS Ubuntu bylo využití procesoru 100 % pouze při spouštění prvků, a poté dochází ke snížení využití procesoru. Ve OS Windows 10 se toto snížení nekonalo a využití procesoru razantním způsobem znemožňuje tvoření větších topologií na dané hardwarové konfiguraci. Pod operačním systémem Windows 10 je vhodné tvořit topologie, které budou obsahovat maximálně 30 směrovačů Mikrotik, kdy dochází k 70 % využití procesoru a k 23 % využití paměti RAM, což znamená využití 7,23 GB z 31,6 GB. OS Ubuntu ve světle výsledků popsaných výše představuje lepší variantu pro tvoření, simulování topologií v programu GNS3, proto je pro další simulace vybrán právě tento operační systém.

GNS3 v prostředí OS Windows 10		
Počet aktivních prvků	**Využití CPU (%)**	**Využití RAM (%)**
10	32	13
20	62	18
30	70	23
40	100	28

Tab. 5.2: Využití procesoru a paměti RAM v operačním systému Windows 10

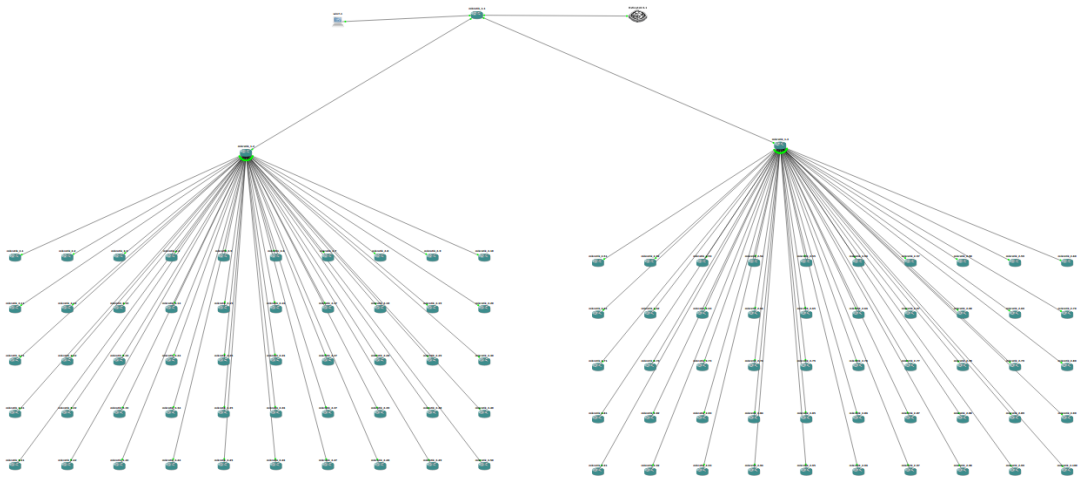


Obr. 5.3: Závislost počtu aktivních prvků na využití CPU (%) a RAM (%) pro OS Windows 10

5.3 Topologie sítě

Topologie vytvořená v prostředí programu GNS3 je znázorněna na obrázku 5.4. V rámci této sítě jsou zařazeny aktivní prvky firmy Mikrotik s operačním systémem RouterOS. Síť obsahuje 100 směrovačů, které představují koncové aktivní prvky. Tyto směrovače obsahují jedno aktivní, připojené rozhraní 100BASE-TX, které umožňuje propustnost 100 Mbit/s, dále jsou součástí topologie dva distribuční směrovače, které jsou propojeny s koncovými směrovači přes rozhraní 100BASE-TX. Nad těmito distribučními směrovači se nachází jeden hlavní směrovač, který je propojen s distribučními směrovači přes rozhraní 1000BASE-T, toto rozhraní umožňuje

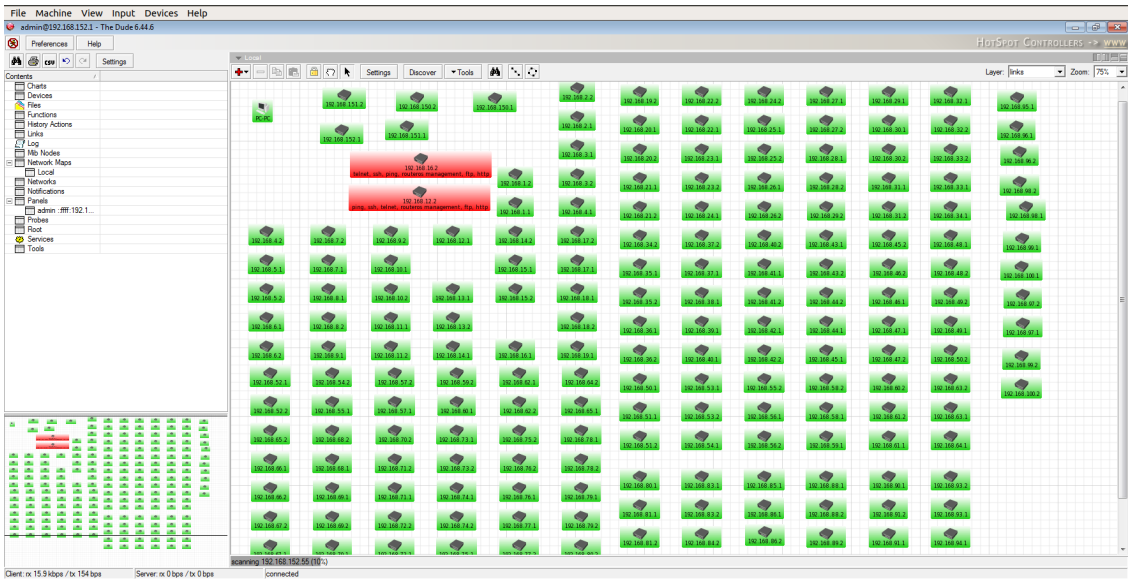
propustnost 1000 Mbit/s.



Obr. 5.4: Topologie sítě

5.3.1 The Dude

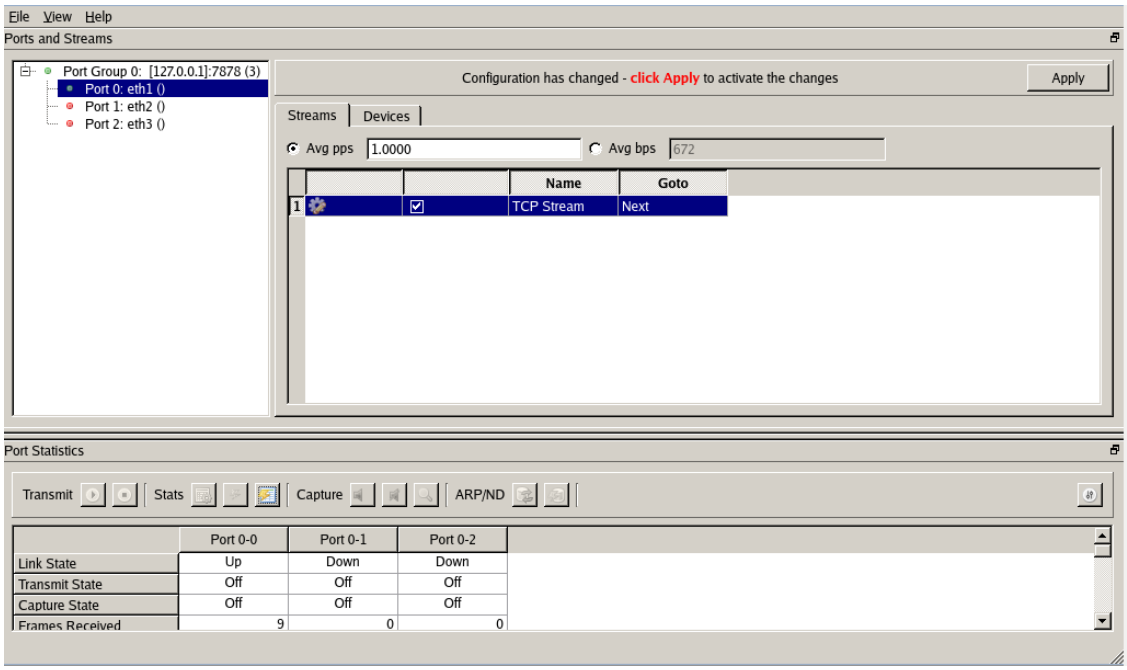
Dále je součástí topologie virtuální stanice s operačním systémem Windows 7 a na této virtuální stanici je zprovozněný program centrální správy The Dude. The Dude je bezplatný program společnosti MikroTik, která může výrazně zlepšit způsob správy síťového prostředí. Automaticky prohledá všechna zařízení v zadaných podsítích, nakreslí a zmapuje sítě, sleduje služby ze zařízení a provádí akce na základě změn stavu zařízení. Zařízení lze nejen monitorovat, ale také je spravovat. Hromadně lze provést upgrade zařízení RouterOS, možnost konfigurace přímo z rozhraní The Dude, možnost spuštění nástroje pro monitorování sítě, atd. Pro zjistitelnost směrovačů v sítí je nutné na všech směrovačích povolit tento program jednoduchým příkazem. V opačném případě dané prvky sítě nebudou zjistitelné, a tím tedy nebudou zařazeny do výsledné topologie vygenerované programem The Dude a nebude tak umožněna centrální správa daného směrovače. Obrázek 5.5 znázorňuje grafické rozhraní programu The Dude.



Obr. 5.5: Grafické rozhraní programu The Dude

5.3.2 Ostinato

V poslední řadě je součástí topologie síťový generátor provozu Ostinato Packet Generator. Ostinato je generátor paketů a generátor síťového provozu s intuitivním grafickým rozhraním a podporou pro automatizaci sítě pomocí výkonného rozhraní API Python. Generátor umožňuje vytvářet a odesílat pakety několika toků s různými protokoly různou rychlostí. Program Ostinato je podporován Windows OS, ale je funkční také na platformách OS Linuxu. Program podporuje standardní protokoly jako jsou Ethernet/802.3/LLC SNAP, VLAN, IP, ARP, TCP, UDP, ICMP a další protokoly různých vrstev síťového modelu OSI. Program obsahuje okno statistik, které běží v reálném čase, a nesou informace o generovaných tocích a jednotlivých paketech, navíc lze tyto pakety zachytávat a následně i zobrazit, například pomocí programu Wireshark. Na obrázku 5.6 je zobrazeno rozhraní programu Ostinato.



Obr. 5.6: Grafické rozhraní programu Ostinato

5.4 Postup při vytváření topologie sítě

Vytváření topologie sítě pro simulace probíhala v tomto pořadí. V prvním kroku proběhlo připravení všech integrovaných aplikací do GNS3. Jednalo se o nainstalování virtualizačního softwaru Qemu a dále pak předat informaci programu GNS3, kde v paměti nalezne spouštěč Qemu. Dále bylo nutné nainstalovat program pro spouštění virtuální stanice Windows 7 na které je umístěn program The Dude. Pro tuto virtuální stanici byl nainstalován program VirtualBox, také v tomto případě bylo nutné v nastavení GNS3 uložit cestu pro spuštění tohoto softwaru. Poslední přidaný program do GNS3 byl generátor provozu Ostinato. Tento program se dá stáhnout z oficiálních webových stránek GNS3. Výhodou stažení tohoto zařízení přímo ze stránek GNS3 je v před-připravenosti virtuálního zařízení pro práci v GNS3, což znamená, že se pouze přidá dané zařízení do knihovny zařízení a generátor provozu Ostinato je připraven k použití. Po nastavení programu GNS3 a všech externích aplikací je možné začít tvořit topologie z zařízení obsažených v knihovně. Samotná simulace vznikala tímto způsobem. Nejdříve se vytvořil projekt v programu GNS3, při vytváření projektu se musí určit, jakým způsobem bude fungovat GNS3 server, na kterém se následně budou tvořit prvky, spoje, atd. V našem případě byla vybrána možnost, která provozuje server na lokálním počítači bez virtuální stanice serveru, a to z důvodu dostatečného výkonu této varianty pro účely naší topologie. Následně byly do topologie vloženy směrovače, kterých v celé topologii je 103. Hi-

erarchie směrovačů vychází z reálné, existující topologie. Jednotlivé prvky se mezi sebou propojily podle požadované přenosové technologie a rychlosti. Následně byla na všech směrovačích provedena základní konfigurace, ve které se nastavily adresy všech použitých rozhraní na směrovačích. Topologie je nakonfigurována do stavu, kdy všechny směrovače dokážou mezi sebou komunikovat, a to díky nastavenému dynamickému směrovacímu protokolu Routing Information Protocol (RIP). Komunikace mezi všemi směrovači se ověřila příkazem *ping* na konkrétní adresy. V topologii jsou využity privátní adresy 192.168.X.X/24. Maska sítě byla nastavena na /24, což dovoluje přiřadit 254 uzlů každé sítí, a měla by představovat zcela dostatečný adresní prostor. Dále byla do topologie vložena virtuální stanice obsahující program centrální správy The Dude. Po spuštění této virtuální stanice a po následném načtení operačního systému se spustil program The Dude. Tento hned po startu nabízí domovskou stránku, na které lze zadat rozsah podsítí, pro které má program provést skenování. Toto skenování může probíhat dvěma způsoby. První způsob představuje rychlou variantu, při které se používá příkaz *ping* na adresy ze zadaného rozsahu podsítí. Druhý způsob představuje spolehlivé skenování podsítí, kdy se skenuje všechny služby, které se mohou v síti vyskytovat. Tato metoda je značně pomalejší oproti rychlému skenování. Pro účely této simulace postačil rychlý sken podsíti, a to z důvodu využívání směrovačů, které podporují *ping*, který patří do protokolu ICMP. Poslední zařízení, které se přidalo do topologie, byl generátor provozu Ostinato.

6 GENERÁTOR SÍŤOVÉHO PROVOZU

Ve vytvořené topologii se zprovoznila komunikace mezi všemi uzly, což umožnilo pokračovat k dalšímu bodu zadání, a to vytvoření generátoru síťového provozu. Jako předloha dat, které se měly generovat, byly použity naměřené hodnoty provozu z části reálné přístupové sítě. Tyto hodnoty byly náměrem jednoho měsíce datového provozu v sítí na přístupové síti, tvořené z Mikrotik směrovačů a přenosová technologie byla v topologii zastoupená Gigabit Ethernetem (1000 Mbit/s) a Fast Ethernetem (100 Mbit/s).

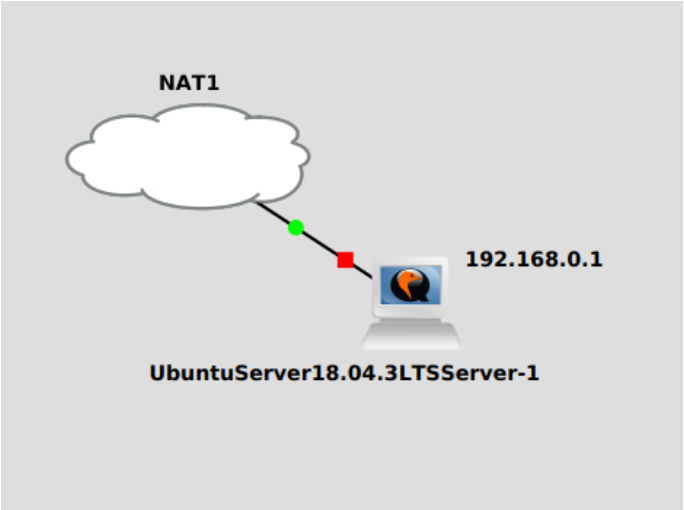
Tento náměr v sobě obsahoval časové razítko, označení rozhraní, obsahoval naměřenou hodnotu přenesených dat. Tato data byla v jednotkách bit/s. K tomu, aby šlo data vhodně použít, bylo nutné provést korekční přepočty naměřených dat tak, aby odpovídala plánované části segmentu přístupové sítě. Náměr obsahoval hodnoty Downloadu, tak i Uploadu. Pro simulace byl z těchto náměru vybrán 3 hodinový úsek datového provozu v sítí a tento úsek byl použit jako předloha datového provozu pro síťový generátor. V programu GNS3 bylo vyzkoušeno více možností různých kombinací a pro prvotní ověření byla zvolena topologie s 20 koncovými stanicemi. V této ověřovací topologii, která obsahuje jeden server a 20 stanic, bylo třeba přerozdělit jednu hodnotu download náměru rovnoměrně na 20 stanic, takže se hodnoty 3. sloupce vydělily 20. Tak bylo zajištěno, že download byl rovnoměrně rozložen mezi koncové stanice. Totéž se provedlo s hodnotami uploadu. Každá koncová stanice pošle 1/20 z celkového uploadu na server. Dvacet stanic tímto dosáhne celkové hodnoty uploadu.

Před samotným tvořením nástroje pro generování provozu se ještě musela provést změna v topologii. Konkrétně byla topologie rozšířena o server, který představoval virtuální počítač, obsahující operační systém Ubuntu 18.04. Tento server má přidělené 4 virtuální jádra procesoru a také 2048 MB paměti RAM. Dále je na stanici nastaven virtuální síťový adaptér Intel Gigabit Ethernet (e1000). Následně byla topologie rozšířená o 20 koncových stanic. Taktéž obsahují operační systém Ubuntu 18.04. Paměť RAM je u každé stanice nastavena na 800 MB a procesor obsahuje dva virtuální jádra.

Na všech stanicích se nejdříve nastavilo síťové rozhraní. Na rozhraní se deaktivoval protokol IPv6, jelikož topologie je nastavena protokolem IPv4. Podle zapojení v sítí se přiřadila IP adresa a maska sítě. Dále se na všech stanicích povolily porty v bráně firewall. V systému Ubuntu se toto nastavení provádí pomocí příkazu *sudo ufw*. Příkaz lze provádět jedině jako root uživatel. Nativně je firewall stanic Ubuntu nastaven tak, aby povoloval výstupní komunikaci, a vstupní komunikaci zamítá. Tímto způsobem se počítač chrání před možnými útoky. Na serveru se povolily porty 12385-12404 pro komunikaci TCP. Tímto se zajistí, aby všech 20 klientských

stanic mohlo přistupovat se svou TCP komunikací na svůj vlastní port. Tak stejně se uvolnilo 20 portů pro komunikaci UDP. Jsou to porty 12405-12424. Na klientských stanicích se povolily dva porty. Vždy tak, aby každá stanice měla povolen přístup na jeden port pro TCP komunikaci a jeden port pro UDP komunikaci. Na stanicích se tedy povolilo dohromady 40 portů. Pro TCP to jsou porty 12345-12364 a pro UDP 12365-12384.

V další fázi bylo vhodné na stanicích doinstalovat aplikace pro jednoduché testování sítě. Jsou to aplikace *iperf* a *ifconfig*. V tomto případě bylo nutné, aby se stanice mohly připojit k internetu a dané balíčky aplikace stáhnout. K tomu, aby se stanice uvnitř topologie GNS3 mohla připojit k internetu, slouží uzel NAT. Díky tomuto uzlu není topologie GNS3 přímo přístupná z internetu nebo místní sítě LAN. Pokud by toto bylo nutné, měl by se použít uzel Cloud. Uzel NAT je velice komfortní z hlediska uživatelského využití. Tento uzel NAT vyžaduje buď GNS3 VM, nebo počítač s Linuxem s nainstalovaným libvirt, což je open-source API a nástroj pro správu virtualizace platformy. Může být použit pro správu KVM, Xen, VMware ESXi, QEMU a dalších virtualizačních technologií. Libvirt je nezbytný pro vytvoření rozhraní virbr0, aby tento uzel fungoval. Operační systém Ubuntu tento nástroj obsahuje. Ve výchozím nastavení uzel NAT provozuje server DHCP s předdefinovanou oblastí v rozsahu 192.168.122.0/24. V našem případě postačí, když uzel připojíme ke stanici Ubuntu, a dále se pouze zkontrolovalo, zda je na daném rozhraní povolen protokol DHCP. Zapojení NAT k stanici je zobrazen na obrázku 6.1.



Obr. 6.1: Zapojení uzlu NAT ke stanici

6.1 Skript pro vysílání datového provozu

V této části je popsán skript, který byl použit pro generování síťového provozu v topologii. Skripty jsou součástí přílohy, ale jeho určité části jsou popsány zde. Pro tento skript byl vybrán skriptovací nástroj *Bash script*. Tento jednoduchý nástroj sekvenčně provádí příkazy.

Nejdříve je v skriptu hlavička *#!/bin/bash*, která při spuštění informuje shell, že má spustit */bin/bash* interpretr. Další část obsahuje deklaraci pole *my_array*. Do tohoto pole se vložila připravena data, která byla vyexportována z náměru reálné sítě. Jednotlivé hodnoty v poli jsou odděleny mezerami. Hlavní část kódu se liší podle toho, jaký typ provozu má být generován. V rámci topologie budou provedeny dvě simulace. První simulace bude obsahovat generátor, který bude vysílat pouze TCP provoz, a to jak upload, tak i download. Druhá simulace bude generovat UDP provoz.

Generátor TCP provozu (ze serveru ke klientům – Download) obsahuje tuto hlavní část programu:

```
for i in "${my_array[@]}"
 do
     declare -i k=12385
    for j in {202..221}
      do
dd if=/dev/zero bs=$i count=1K |nc -vnn 192.168.$j.2 $k -w1
         echo "$i, $j, $k"
         k=$k+1
    done
sleep 30
 done
```

Program zde vstupuje do smyčky *for i in "${my_array[@]}"*, která postupně prochází polem s hodnotami, které představují velikost souborů, které se budou přenášet. Dále je deklarována proměnná *k*, které je přiřazená hodnota 12385. Tato proměnná obsahuje číslo portu. Dále je zde obsažena další smyčka *for j in {202..221}*, ta pracuje s hodnotami od 202 až 221, kde tato čísla jsou pak využita pro nastavení doručovací adresy sítě. Skript má smyčku ve smyčce, a to znamená, že pro jednu hodnotu z pole *my_array* se provede pro všechny buňky druhého pole s adresami, a až poté se přesune program k další buňce z pole *my_array*. Další řádek obsahuje spojené dva příkazy do jednoho pomocí Unixové roury neboli *pipe*. První příkaz je příkaz *dd*. Tento příkaz se běžně používá pro převod či kopírování souborů. Tento příkaz může také číst anebo zapisovat do souboru. Běžné použití je při tvoření záloh pevného disku, nebo také generování náhodných dat o určitě stanovené velikosti.

První parametr příkazu je stanovení cesty k souboru. V tomto případě je cesta tato */dev/zero*. Tato cesta se odkazuje na speciální soubor v unixovém operačním systému, který generuje nulové znaky. Tato funkce umožňuje vytvořit soubory požadované velikostí, aniž bychom potřebovali jakékoliv místo na pevném disku. Dále je zde parametr *bs=$i*, ten určuje velikost souboru a je mu přiřazená proměnná *$i*, která odkazuje na hodnotu v poli na aktuální adrese v poli v B/s. A poslední parametr příkazu *dd* je parametr *count=1K*. Ten určuje počet bloků.

Dále je deklarována proměnná *k*, které je přiřazená hodnota 12385. Tato proměnná určuje číslo portu. Příkaz, který je spojen rourou k příkazu *dd* je příkaz *nc*. Jedná se o obslužný program, který podporuje širokou škálu možností pro správu sítí a sledování toku provozních dat mezi systémy. Program se také používá pro ladění a průzkum sítě, který obsahuje mnoho funkcí, které umožňují vytvořit různé typy připojení. V našem případě je použit k tomu, aby vytvořil TCP, nebo UDP spojení na požadované adrese a portu, a na tu následně poslal určité množství dat. Příkaz vypadá takto: *nc -vnn 192.168.$j.2 $k -w1*. První parametr *-v* způsobí, že se v konzole vypisují zprávy o průběhu. Jsou to informace o navázaní spojení, vytvoření relace, průběh relace, stav relace a po ukončení relace vypíše celkové množství přenesených dat, délku trvání relace a rychlost, jakou byla data přenesena v jednotkách B/s. Další parametr *-nn* říká příkazu, aby neprováděl DNS vyhledávání. Dále je příkazu přiřazená IP adresa, která je ve tvaru 192.168.X.2. Kde X určuje, na kterou z 20 koncových stanic relace bude směrovat. Další parametr je číslo portu a to se rovná proměnné *k*. Toto číslo se inkrementuje o hodnotu jedna s každým dalším cyklem smyčky pro adresy. Po ukončení druhé smyčky se znovu hodnota portu resetuje na počáteční hodnotu. Poslední parametr příkazu *nc* je parametr *-w1*. Tento parametr určuje timeout pro spojení. Terminál čeká N sekund po ukončení standardního vstupu. Příkaz tímto počká na vytvoření relace. Pokud nedojde ke spojení, příkaz se ukončí. Pro kontrolu správnosti portů, adres a velikostí souborů se do konzole vypisují právě tyto hodnoty. Slouží k tomu příkaz *echo "$i, $j, $k"*. Posledním příkazem je příkaz *sleep 30*. Stejně jak byly hodnoty náměru z reálné sítě co 30 sekund, je tato vlastnost zachována i v tomto skriptu. Funkčně to znamená, že po ukončení jednoho bloku program počká 30 sekund, než se přesune k dalšímu cyklu smyčky s novou velikostí souboru.

Generátor TCP v opačném směru to znamená od klientských stanic na server – Upload se liší od TCP – Download generátoru v dále popsaném. Druhá smyčka, která byla dříve použitá pro určování adres, je zde využita ke změně portů, jelikož adresa, na kterou má být provoz směřován je vždy stejná, a to 192.168.200.2. Tuto smyčku se tím pádem využilo pro změnu portů. Hlavní část kódu je zobrazena níže.

```
for i in "${my_array[@]}"
```

```
do
   for j in 12364
         do
     dd if=/dev/zero bs=$i count=1K |nc -vnn 192.168.200.2 $j -w1
     echo "$i, $j"
          done
       sleep 30
done
```

Simulace pro UDP provoz se liší ve skriptech v jednom parametru příkazu *nc* a konkrétně se jedná o parametr *-u*, který určuje použití UDP protokolu, namísto TCP, a také se liší v číslech portů.

6.2 Skript pro naslouchání na straně přijímače

Protokol TCP vyžaduje na straně přijímače ke svému správnému fungování nástroj, který bude naslouchat na daném portu, a bude vysílači odpovídat na signalizační zprávy. V našem případě se vytvořil skript, který po svém spuštění provede kód, který je zobrazen níže.

```
gnome-terminal --tab -- bash -ic
"for i in {1..100000}; do nc -vvlnp 12345 >/dev/null; done"
```

Tento kód nejdříve provede příkaz *gnome-terminal –tab – bash -ic*. Ten otevře terminál stanice Ubuntu, ale otevře se jako záložka terminálu, jelikož na straně serveru je 20 přijímačů TCP proudů dat, a tím pádem je potřeba, aby daný skript spustil naslouchání na všech požadovaných portech. Takovým způsobem se otevře 20 záložek terminálu a každý provede svůj vlastní kód, a nebude se otevírat 20 oken terminálu. Následně skript vstoupí do smyčky typu *for i in {1..100000};*, která zaručí, že kód uvnitř smyčky se provede 10000 krát.

V dalším kroku vstupuje program do smyčky a zde se nachází příkaz *nc*. Netcat (nebo nc) je obslužný program příkazového řádku, který čte a zapisuje data přes síťová připojení pomocí protokolů TCP nebo UDP. Netcat je multiplatformní a je k dispozici pro systémy Linux, MacOS, Windows a BSD. Pomocí Netcat lze ladit a sledovat síťová připojení, vyhledávat otevřené porty, přenášet data jako proxy a další. Balíček Netcat je předinstalován na MacOS a populárních linuxových distribucích jako Ubuntu, Debian nebo CentOS. V tomto případě je tento příkaz použit v režimu naslouchání. Režim naslouchání je zajištěn díky přidanému parametru *-l*, který spustí obslužný program v režimu naslouchání. Dále je v příkazu zadán parametr *-vv*. Tento parametr říká obslužnému programu, aby do terminálu vypisoval detaily

přenosu a relace. Je zde možnost zadat pouze parametr *-v*, ten by také vypisoval detaily přenosu a relace, ale v omezeném množství. V případě testování a simulování bylo vhodnější, aby terminál vypisoval co nejvíce informací pro případné opravování chyb, nebo pro větší kontrolou nad tím, co se na trase děje. Další parametr zadaný v příkazu je parametr *-n*. Tímto příkazem říkáme obslužnému programu, aby neprováděl vyhledávání DNS na názvech strojů na druhé komunikující straně. Poslední parametr *-p 12345* dovoluje zadat port, na kterém má stanice naslouchat. V tomto případě stanice bude naslouchat na portu číslo 12345.

Za příkazem *nc* je zadaná cesta, kde se mají ukládat data, která byla doručena. Cesta představuje virtuální soubor v zařízení, kdy utility mohou požadovat data z tohoto druhu zdroje a operační systém je dodává. Ale namísto čtení z disku operační systém generuje tato data dynamicky. V jednoduchosti to znamená, že není potřeba disponovat velkým množstvím pevné paměti na data. Data, která se doručí do stanice, budou přijata, ale nebudou zabírat žádné místo na disku. Po ukončení naslouchání dané relace skript začne od počátku provádět stejný kód, dokud se neukončí práce na zařízení a skript se přeruší.

Níže je zobrazen příkaz pro naslouchání a příjem UDP proudů dat. Příkaz *nc* obsahuje parametr *-u*, ten obslužnému programu říká, že naslouchá na UDP portu. Dále je v tomto příkazu obsažen parametr *-w1*. Tento parametr je pauzou na N sekund, po kterou příkaz čeká, než ukončí příkaz *nc*. V tomto případě proběhne pauza na jednu sekundu.

```
gnome-terminal --tab -- bash -ic
"for i in {1..100000}; do nc -vvulp 12405 -w1; done"
```

7 VÝSLEDKY SIMULACÍ

V této kapitole jsou popsány výsledky, kterých se dosáhlo během simulací na vytvořené topologii. K dosažení výsledků bylo během simulace použito dvou síťových nástrojů. První síťový nástroj byl program Wireshark. Tento program slouží jako protokolový analyzátor a paketový detektor. Nejčastěji se tento program používá k ladění a analýze počítačových sítí, vývoji nových komunikačních protokolů. Nejdříve zachytí provoz na vybraném rozhraní a následně umožňuje použít veliké množství analyzátorů nejrůznějších protokolů, formátů a také lze provoz filtrovat, a to pomocí uživatelem definovaných filtrů. Tímto způsobem lze analyzovat i ten nejmenší detail posílaných paketů, jako je například číslo rámce, či časové razítko. Simulační program GNS3 navíc umožňuje nasadit do topologie sondu, kterou lze zařadit přímo na linku, například mezi serverem a směrovačem. Po umístění této sondy, je zachytáván veškerý provoz, který touto linkou prochází, a to v obou směrech. Po ukončení simulace se pouze toto zachytávání ukončí a v souborech projektu lze lokalizovat *.pcap* soubory, které obsahují veškerý provoz v síti.

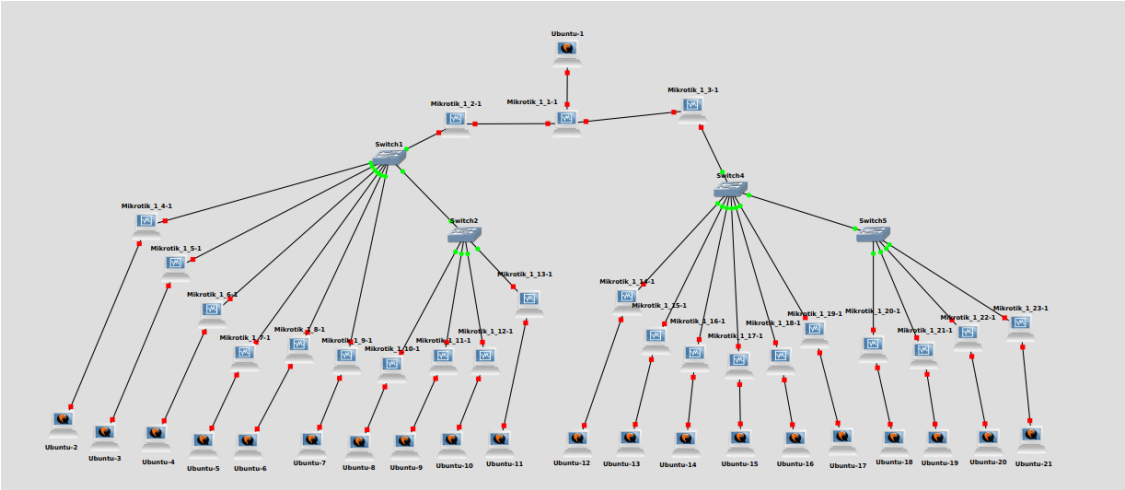
Druhým programem, kterým se snímalo chování sítě byl program Ntopng. Jedná se o počítačový software pro sledování provozu v počítačové síti. Je navržen jako vysoce výkonná náhrada za jeho předchůdce Ntop. Ntopng je open-source software. Binární verze jsou k dispozici pro Unix i Linux platformy. Pro operační systém Windows je k dispozici demo binární verze, která je ovšem omezena na 2000 paketů. Tento program byl v topologii nainstalován na stanici serveru. Mezi hlavní vlastnosti programu patří:

- Třídění síťového provozu podle mnoha kritérií včetně IP adresy, portu, protokolu L7, propustnosti, autonomních systémů (AS).
- Zobrazení síťového provozu v reálném čase a aktivní účastníci.
- Vytváření dlouhodobé zprávy pro několik metrik sítě, včetně propustnosti a aplikačních protokolů.
- Sledování a zobrazování propustnosti, síťové a aplikační latence, Round Trip Time (RTT), statistiky TCP (opakované přenosy, pakety mimo pořadí, ztracené pakety) a přenášené bajty a pakety.
- Analýza IP provozu a řazení podle zdroje nebo cíle.

Topologie obsahuje směrovače Mikrotik se systémem RouterOS. Tyto směrovače byly emulovány pomocí programu QEMU. Při realizaci bylo nalezeno omezení rychlosti 1 MB/s. Toto omezení rychlosti bylo definováno výrobci jako omezení, aby nedocházelo ke zneužití simulačních nástrojů a jejich připojení ke globální síti. Proto v této souvislosti proběhlo zkontrolování nastavení směrovačů, a to na konfiguraci omezení rychlosti v QoS, špatně nastavené rozhraní. Dále proběhla kontrola konco-

vých stanic. A v poslední řadě se zkontroloval skript pro vysílání a příjem TCP či UDP provozu. Problém se v těchto konfiguracích a skriptech nenašel, a proto proběhla kontrola hardwarových prostředků počítače, zda během simulace nedochází k zahlcení CPU a RAM. Toto se také neprokázalo, jelikož po spuštění všech prvků topologie a následně po spuštění simulace využití procesoru nepřesahovalo 40 % a využití RAM nepřesahovalo 85 %. Poslední možnost byla, že omezení rychlosti je způsobeno emulačním softwarem. Proto proběhly testy na rychlost přenosového média. Nejdříve se otestovalo, jakou rychlost dosahuje přenos souboru mezi dvěma stanicemi Ubuntu propojenými přenosovou technologii Gigabit Ethernet. Rychlost se měřila příkazem *iperf*. Na první stanici se přidal k příkazu parametr *-s*, který upřesnil, že daná stanice má být v režimu naslouchání. Na druhé stanici se k příkazu přidal parametr *-c 192.168.1.1*. Tento parametr určil adresu vzdálené stanice. Tímto způsobem se změřila šířka pásma mezi dvěma stanicemi. Ta se rovnala 991 Mbit/s. Tímto se vyloučilo, aby za tak značným omezením stály koncové stanice, které byly virtualizovány programem VirtualBox. Poslední možností, kde mohlo docházet k omezení, byly tedy Mikrotik směrovače emulovány programem QEMU. V zjednodušené verzi topologie s dvěma stanicemi a jedním Mikrotik směrovačem se toto omezení potvrdilo. V souvislosti s tím se musel změnit virtualizační program, který emuloval Mikrotik prvky. Pomocí obrazu CHR Mikrotik směrovače byl vytvořen virtuální stroj programem VirtualBox. Pro vyzkoušení byla opět použitá modifikovaná topologie obsahující dvě koncové stanice Ubuntu a jeden Mikrotik směrovač, ale tentokrát emulován pomocí programu VirtualBox. Výsledek šířky pásma se rovnal 695 Mbit/s. Tato šířka pásma byla postačující pro naše testování. Tímto se ověřilo, že omezení rychlosti na 1 MB/s bylo definováno výrobci pro emulované stroje pomocí QEMU.

V souvislosti s tímto se upravila topologie k podobě znázorněné na obrázku 7.1. Směrovače byly virtualizovány pomocí programu VirtualBox, a z důvodu složitějšího duplikování jednotlivých virtuálních stanic jich bylo do topologie vloženo pro potřeby měření 23. Také byly do topologie zařazeny 4 switche, a to z důvodu omezení maximálního počtu síťových rozhraní virtuálních stanic Mikrotik. Všechny linky jsou



Obr. 7.1: Vybraný segment topologie

7.1 Simulace TCP provozu

V této podkapitole jsou popsány výsledky TCP simulace. Tato simulace trvala 1 hodinu a 36 minut. Množství přenesených dat a paketů na serveru je znázorněno v tabulce 7.1. Poměr uploadu k downloadu je roven 18,8 %. Tato hodnota naznačuje relativně vysoký upload ze strany klientů.

Množství přenesených paketů/dat TCP simulace	
Odesláno:	Přijato:
32 307 099 paketů	10 622 476 paketů
44,1 GB	8,3 GB

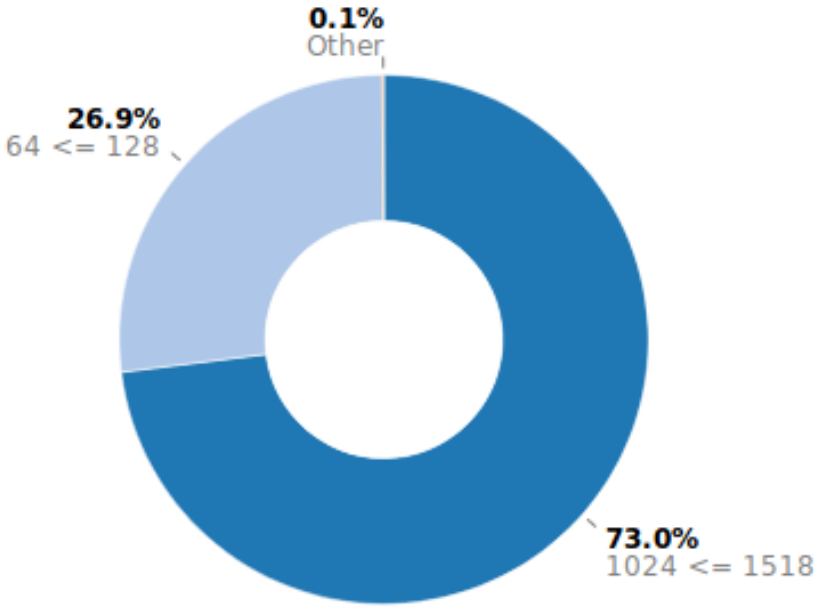
Tab. 7.1: TCP Simulace - Přenesených dat/paketů

Během simulace TCP došlo k posílání paketů pro kontrolu přenosu, kdy došlo znovu k posílání paketů, k přijetí paketů mimo pořadí a také k ztracení paketů. Množství těchto nežádoucích jevů je popsán v tabulce 7.2. Součet všech paketů, které byly v sítí přijímány, nebo odesílány, byl roven 42 929 575 paketů. Pakety pro kontrolu přenosu byly zastoupeny v minimální mezi. Celkem bylo přeneseno 11 679 paketů pro kontrolu přenosu. V poměru k celkovému množství přenesených paketů v síti tvoří 0, 0272 %, což je zanedbatelné číslo. Společnost Mikrotik ve své technické dokumentaci stanovuje ke svým produktům určitou toleranci ztrátovosti. Pro běžný Mikrotik směrovač RB2011UiAS-RMr2 je tato tolerance stanovena na 0,1 % ztrátovosti paketů o velikosti 64 B, 512 B a 1518 B [8].

	Odesláno:	Přijato:
Znovu odeslání	8 paketů	79 paketů
Mimo pořadí	5 462 paketů	601 paketů
Ztracený	5 229 paketů	300 paketů

Tab. 7.2: Retransmissions / Out of order / Lost

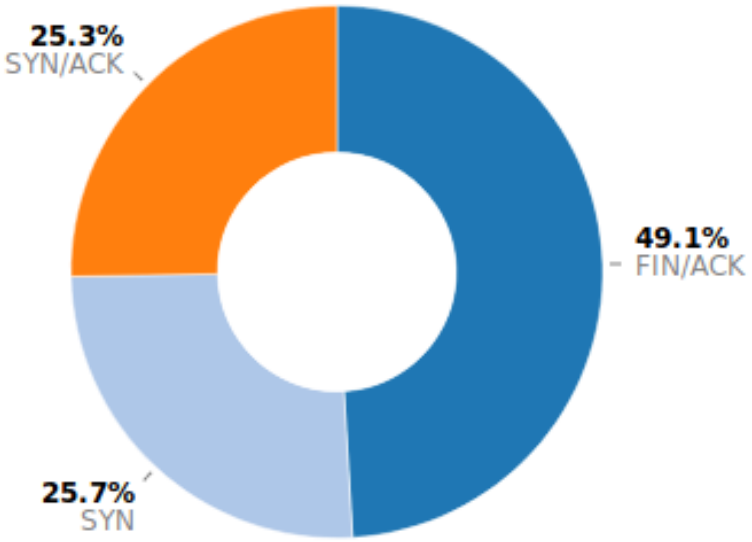
V zachycených paketech obsahujících TCP provoz se očekává, že určité procento paketů bude datových a dále pak signalizačních, s tím, že datových paketů by mělo být procentuálně více. Tento předpoklad se naplnil a výsledné procentuální zastoupení paketů podle délky je znázorněno na obrázku 7.2. Datových paketů o délce 1514 B je v celém vzorku zastoupeno 73 % a pro signalizační pakety o velikosti od 64 B po 128 B je v simulaci zastoupeno 26,9 %.



Obr. 7.2: Délka paketů TCP simulace

Výsečový graf 7.3 zobrazuje poměr signalizačních zpráv protokolu TCP. Nejvíce byly v simulaci zastoupeny zprávy FIN/ACK. Tyto zprávy TCP používá k ukončení relace. Tyto pakety jsou vyžadovány na každém koncovém bodu TCP relace. Signalizační zpráva SYN byla zastoupena 25,7 %. Touto zprávou klient posílá žádost o spojení, a navíc provádí synchronizaci sekvenčního čísla mezi zařízeními. Poslední část grafu tvoří zprávy SYN/ACK, ty představují 25,3 %. Tyto zprávy posílá server

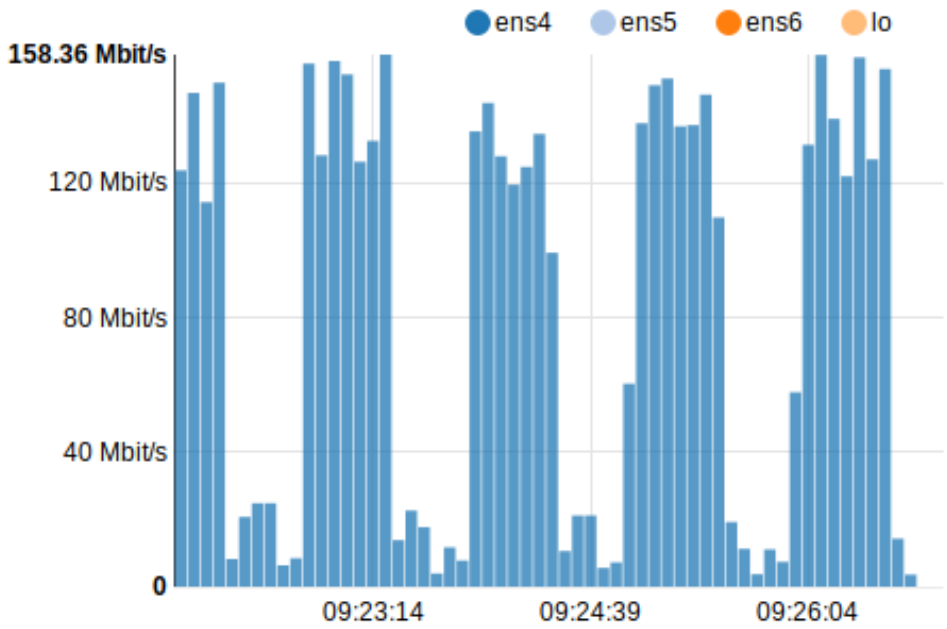
jako odpověď klientovi. SYN zpráva z lokální stanice slouží k iniciování spojení a ACK zpráva k potvrzení předešlého paketu. Spojení těchto dvou zpráv urychluje celý proces, kdy se nemusí zvlášť posílat potvrzovací ACK zprávy.



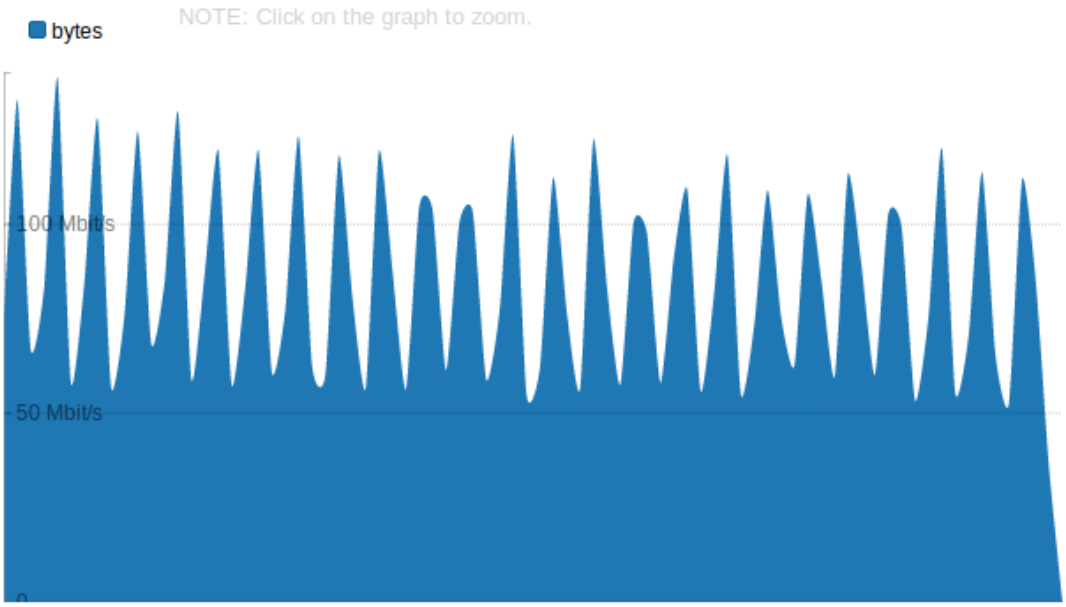
Obr. 7.3: Rozložení TCP příznaků

Na snímku 7.4 je zobrazena přenosová rychlost v reálném čase během simulování TCP provozu. Provoz je zachytáván z rozhraní ens4, který představuje rozhraní, kterým je stanice serveru připojena ke směrovači Mikrotik. Z tohoto grafu lze vyčíst nejvyšší dosaženou rychlost přenosu dat. Tato rychlost je rovná 158,36 Mbit/s. Z grafu lze taktéž vyčíst určité časové úseky, ve kterých přenosová rychlost nedosahuje ani 30 Mbit/s. Tyto úseky odpovídají času, kdy probíhá 30 sekundové okno ve skriptu, ve kterém se čeká, než se začne posílat další blok dat. Skripty na jednotlivých stanicích byly spouštěny postupně, takže některé stanice ještě posílaly data i během tohoto 30 sekundového okna.

Na následujícím obrázku 7.5 je znázorněna závislost času na přenosové rychlosti během celé simulace. Tento obrázek poukazuje na mírný pokles přenosové rychlosti. Tento pokles je způsoben menšími bloky dat, které se s rostoucím časem posílaly. Již při analýze vzorku dat z reálné sítě byl tento pokles v čase patrný.



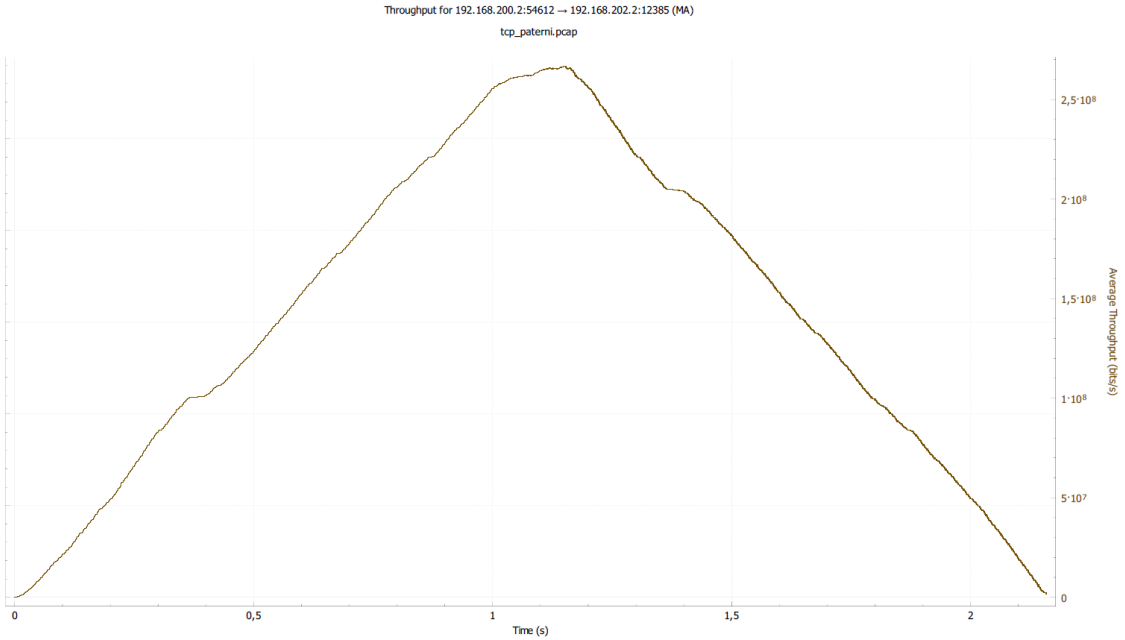
Obr. 7.4: Přenosová rychlost v reálném čase TCP simulace



Obr. 7.5: Přenosová rychlost celé TCP simulace

V grafu 7.6 je zobrazena závislost času na průměrné propustnosti. Tento graf ukazuje tuto závislost pro jeden konkrétní TCP datový proud z adresy 192.168.200.2 a z portu 54612, což je stanice serveru, a datový proud směřuje na klientskou stanici s adresou 192.168.202.2 a portem 12385. Z grafu lze vyčíst, že data byla přenášena

mezi stanicemi po dobu 2,15 s, kdy nejdříve propustnost lineárně roste v čase, a tento úsek grafu poukazuje na činnost vysílací stanice, která začala vysílat datový proud síti. Postupně se propustnost dostane na špičku, ve které využívá maximální možnou propustnost, po přenesení všech dat začne propustnost opět klesat až k dosažení nulové propustnosti, v tomto momentu dojde k ukončení relace a skript pokračuje k dalšímu požadavku ve frontě.



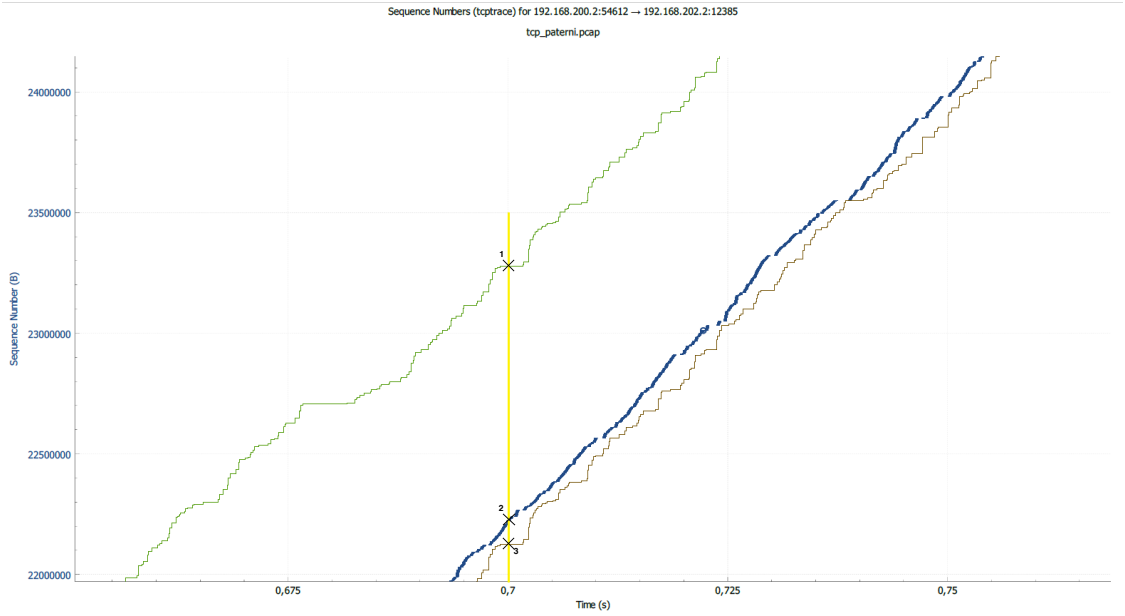
Obr. 7.6: Graf průměrné propustnosti jednoho TCP přenosu

Graf časové sekvence zobrazen na obrázku 7.7 ukazuje datový tok v čase. Osa x představuje čas (s). Pro vysvětlení byla do grafu zařazená žlutá čára protínající osu x v čase 0,7s. Osa y obsahuje pořadová sekvenční čísla TCP. Pořadová sekvenční čísla jsou zástupci odeslaných bajtů. Pořadové číslo se zvyšuje o 1 na každý 1 bajt odeslaných dat TCP. V ideálním případě graf představoval lineární narůst v čase. Strmost průběhu teoreticky představuje šířku pásma vedení. Čím je strmější čára, tím znázorňuje vyšší propustnost. Graf obsahuje tři křivky – zelenou, modrou a hnědou.

Prostřední křivka – modrá, která je označena značkou č.2 představuje TCP segmenty, které se skládají z krátkých paprsků. Čím delší je tento paprsek, tím více přenesených dat na paket.

Křivka pod TCP segmenty – hnědá, označená značkou č.3 představuje ACK pakety z přijímače. Vzdálenost mezi ACK a TCP daty v daném časovém bodě představuje bajty v pohybu. Takže na serveru v našem vybraném místě grafu v čase 0,7 s byl odeslán 22 200 000 TCP bajt a zároveň byl přijat 22 120 000 ACK bajt, pak je v oběhu 80 000 B, které ještě nebyly doručeny.

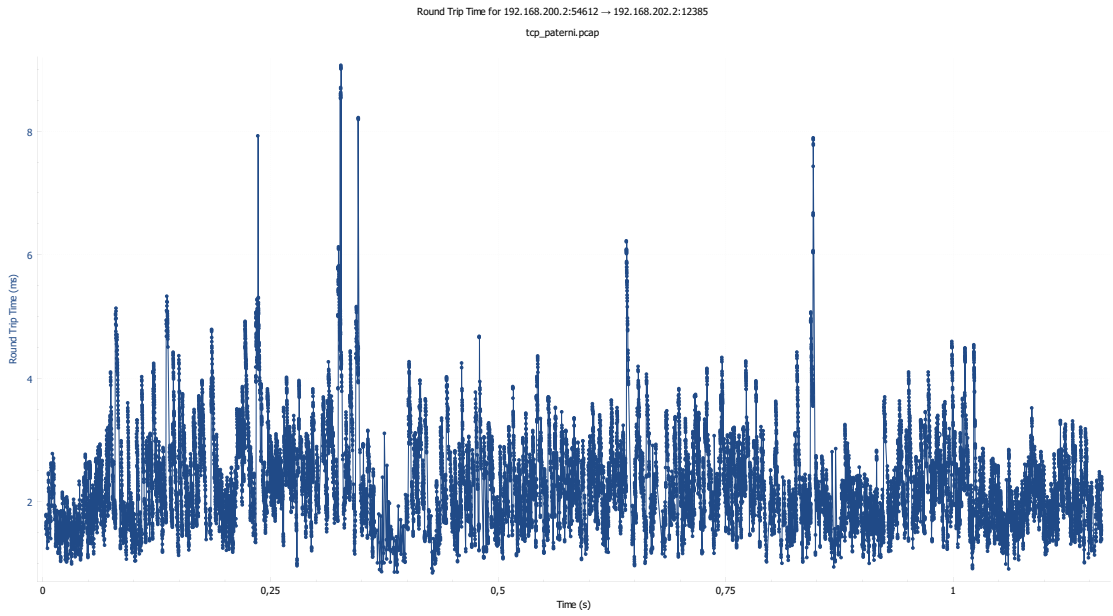
Poslední křivka – zelená zobrazuje vypočítané přijímací okno klienta a je označena značkou č.1. Tento výpočet je dán součtem ACK čísla a aktuálního inzerovaného okna pro příjem. Pokud je aktuální ACK 22 120 000 B a okno inzerovaného příjmu 1 115 000 B, pak bude vypočítané okno příjmu o velikosti 23 275 000 bajtů. Vzdálenost mezi aktuálním sekvenčním číslem TCP (22 200 000 B) a vypočteným oknem pro příjem (23 275 000 B) je velikost bufferu 1 075 000 B. To znamená, že klient dokáže uložit do vyrovnávací paměti 1 075 000 B.



Obr. 7.7: Závislost času na sekvenčním čísle (tcptrace)

Další parametr, který je v této části práce popsán je parametr RTT (Round-trip time). Tento parametr v jednotce ms udává dobu, která trvá, než síťový požadavek přejde ze startovního bodu do cíle a znovu zpátky do startovního bodu. Parametr RTT je důležitá metrika při určování stavu připojení v místní síti. Tento parametr je běžně využíván k diagnostice rychlosti a spolehlivosti síťového připojení.

Na obrázku 7.8 je zobrazena závislost času (s) na RTT (ms). Tato závislost je znázorněna pro jeden konkrétní datový tok ze stanice serveru, který má adresu 192.168.200.2 a port 54612. Tento datový tok směřuje na klientskou stanici, který má adresu 192.168.202.2 a port 12385. Pro přesné odečtení z grafu RTT v čase byl použit nástroj, který ze zachyceného provozu vyexportoval hodnoty RTT. Díky tomuto kroku bylo mnohem snazší pomocí funkcí programu Excel určit maximum a minimum hodnoty RTT, a následně i rozptyl. Pro tento konkrétní TCP datový tok se hodnota RTT maximálně rovnala 9,023 ms. Minimální hodnota byla 0,848 ms. Rozptyl hodnot RTT byl roven 0,634 us.



Obr. 7.8: Vývoj zpoždění při přenosu

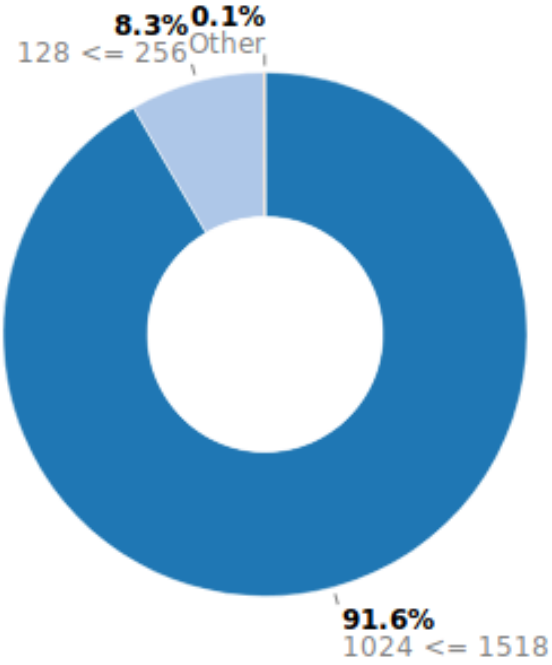
7.2 Simulace UDP provozu

V této podkapitole jsou popsány výsledky UDP simulace. Tato simulace trvala 1 hodinu 19 minut. Během této doby bylo přeneseno určité množství dat, které je znázorněno v tabulce 7.3. Během simulace došlo k zahození celkem 68 paketů, což je zanedbatelné množství pro UDP provoz. Poměr uploadu k downloadu je roven 19 %, což je hodnota velice podobná k výsledku TCP simulace, kde byl tento poměr roven 18,8 %.

Množství přenesených paketů/dat UDP simulace	
Odesláno:	Přijato:
32 516 250 paketů	6 236 214 paketů
43,1 GB	8,3 GB

Tab. 7.3: UDP Simulace - Přenesených dat/paketů

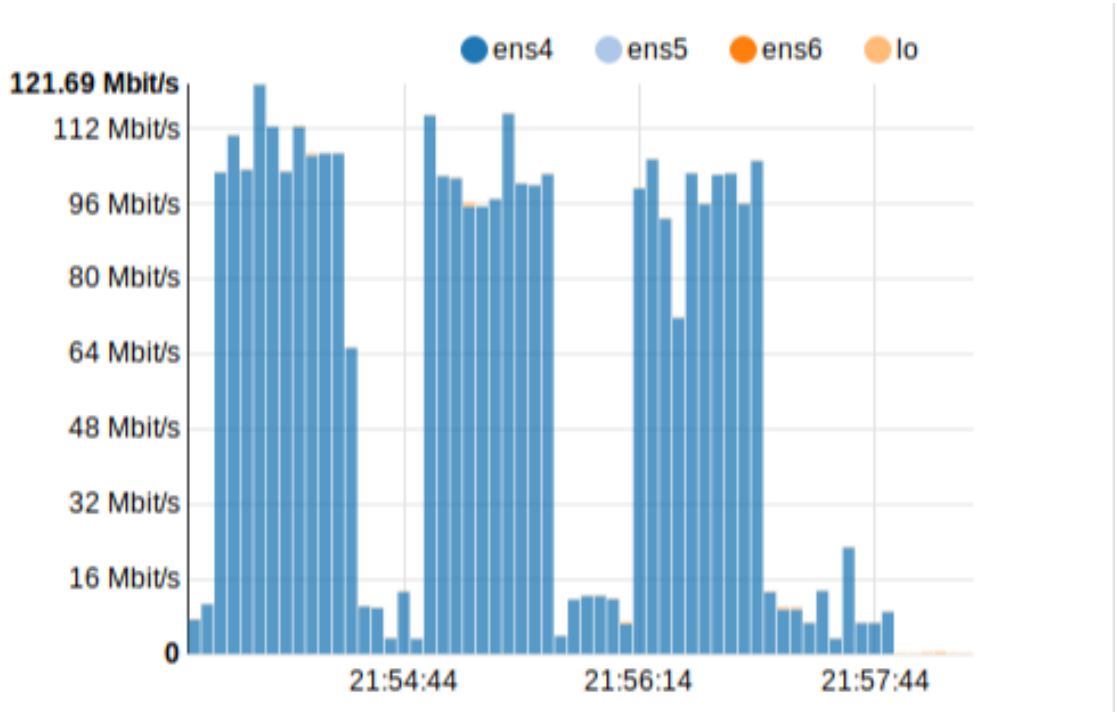
Během simulace bylo odesláno 91,6 % paketů o velikosti 1024 B až 1518 B, pro protokol UDP byla tato velikost rovná 1514 B, dále 8,3 % paketů o velikosti 128 B až 256 B. Tato velikost 146 B odpovídá ethernetovému rámci a tyto pakety byly přítomný za určitým množstvím fragmentovaných paketů UDP, a v tomto paketu docházelo k znovu sestavení bloku. Ve fragmentovaných paketech byla uložena informace o offsetu, který s každým paketem rostl o jednotku 1480 B až do hodnoty 16384 B, což je poslední paket bloku, který měl délku 146 B.



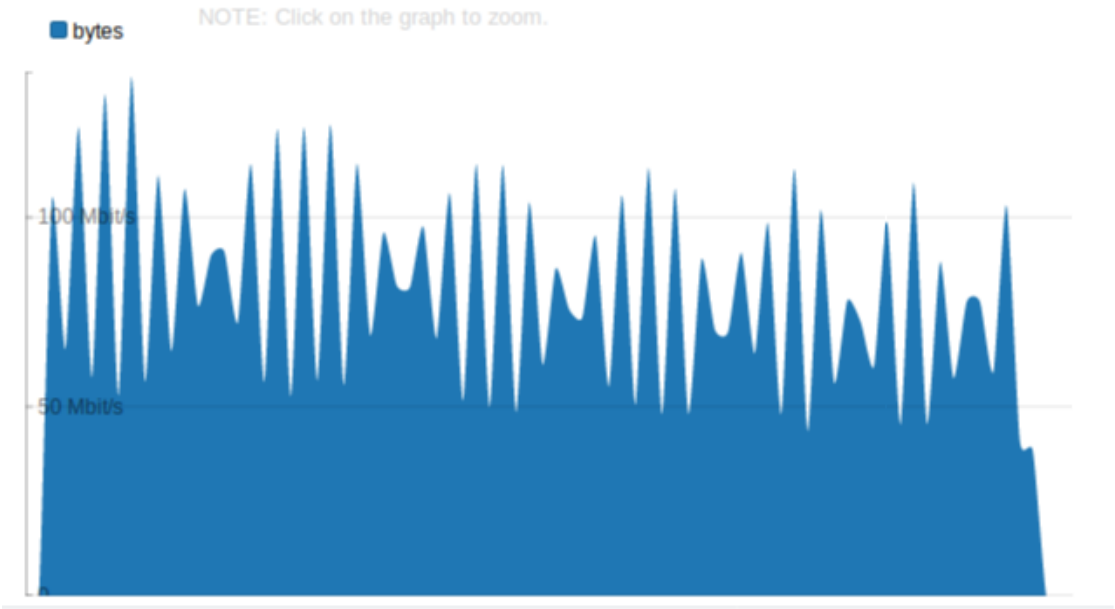
Obr. 7.9: Délka paketů UDP simulace

Graf 7.10 zobrazuje rychlost přenosu v reálném čase. Provoz byl zachytáván na rozhraní ens4. Maximální rychlost dosažená na časové ose byl roven 121,69 Mbit/s. Stejně jak u simulace TCP, jsou v grafu viditelné špičky, kdy docházelo k většímu využití přenosové šířky pásma. Znovu tato okna představují 30 sekundová okna, kdy rychlost se pohybovala okolo 16 Mbit/s.

Na dalším grafu 7.11 je zobrazena přenosová rychlost v průběhu celé simulace.



Obr. 7.10: Přenosová rychlost UDP simulace v reálném čase



Obr. 7.11: Přenosová rychlost celé UDP simulace

8 ZÁVĚR

Tato práce se zabývala programem Graphical Network Simulator 3 a možnostmi, které tento program nabízí. V teoretické části byly popsány základní vlastnosti programu GNS3 i s možnými integracemi programu jako jsou například různé emulátory. Dále se práce věnovala základním vlastnostem směrovačů společnosti Mikrotik s RouterOS s jejich základními funkcemi. Následně práce popisovala základy směrování v IP sítích. V poslední teoretické části byly popsány typy doručování paketů, které jsou běžné v IP sítích.

Praktická část se věnovala nejdříve srovnáním operačních systému Windows 10 a Ubuntu pod úhlem výkonu a propojení hardwaru a softwaru GNS3. V tomto srovnání vyšel operační systém Ubuntu jako vhodnější operační systém, a to z důvodu možnosti vytvářet topologie obsahující větší množství spuštěných aktivních prvků. V systému Ubuntu se podařilo spolehlivě odsimulovat 275 spuštěných a funkčních směrovačů Mikrotik. V systému Windows 10 to bylo maximálně 40 spuštěných směrovačů Mikrotik.

V další části práce byla popsána veškerá nastavení, která byla nutná provést pro vytvoření požadované topologie sítě. V poslední části kapitoly byl přiblížen postup při vytváření topologie sítě. S konfiguracemi, které byly provedeny na jednotlivých směrovačích, aby se dosáhlo vzájemné komunikace v rámci sítě.

Práce dále popisuje vytvořený generátor síťového provozu. Jako předloha dat pro generování byly použity náměry z části reálné přístupové sítě. Proběhlo odzkoušení různých kombinací. Pro prvotní ověření byla zvolena topologie s 20 koncovými stanicemi, proto se data generátoru přizpůsobila 20 klientským stanicím. Generátor byl vytvořen pomocí Bash skriptu. Skripty byly uzpůsobeny pro přenos dat v obou směrech (download i upload), a také pro přenosové protokoly TCP a UDP. Nástroj generátoru je přenositelný a lze v nich nastavit adresy a porty koncových stanic. Také je možno v poli dat měnit hodnoty i jejich množství. Příkaz *dd* má své omezení v maximální velikosti tak zvané nulové paměti. V případě naší testovací simulace velikost generovaných dat nedosahovala tohoto omezení.

V následující části práce byly popsány výsledky simulace pro TCP a UDP datový provoz. Datový provoz byl zachytáván a následně analyzován pomoci programu Wireshark, a také pomocí monitorovacího programu Ntopng. Délka simulací a množství přenesených dat byla přibližně podobná.

LITERATURA

[1] GNS3 | The software that empowers network professionals. *GNS3 | The software that empowers network professionals.* [online]. [cit. 2017-12-01]. Dostupné z: https://gns3.com/

[2] WELSCH, Chris. *GNS3 Network Simulation Guide*. 2013. Packt Publishing Limited. ISBN 9781782160809.

[3] CHERNOV, A. Yu. a A. S. KONOPLEV. The use of virtualization technology in the dynamic analysis of software code. *Automatic Control and Computer Sciences* [online]. 2015, **49**(8), 834-837 [cit. 2019-11-20]. DOI: 10.3103/S0146411615080234. ISSN 0146-4116. Dostupné z: http://link.springer.com/10.3103/S0146411615080234

[4] TOLSTOY, Alexander. Qemu. *Linux Format* [online]. Bath: Future Publishing, 2016, (208), 54 [cit. 2019-11-18]. ISSN 14704234. Dostupné z: http://search.proquest.com/docview/1790476880/

[5] *QEMU* [online]. [cit. 2019-11-18]. Dostupné z: https://www.qemu.org/documentation/

[6] *VMware* [online]. [cit. 2019-11-18]. Dostupné z: https://www.vmware.com/support/pubs/

[7] *VirtualBox* [online]. [cit. 2019-11-18]. Dostupné z: https://www.virtualbox.org/wiki/Documentation

[8] *Mikrotik Wiki* [online]. 2011 [cit. 2019-12-16]. Dostupné z: https://wiki.mikrotik.com/wiki/Manual:TOC

[9] BURGESS, Dennis. *Learn RouterOS*. Lexington]: Dennis Burgess, 2009, 391 s. : il. ISBN 978-0-557-09271-0.

[10] *Journal of computing sciences in colleges*. Belmont, NC: Consortium for Computing Sciences in Colleges, 2006. ISSN 1937-4771.

[11] PUŽMANOVÁ, Rita. *Routing and switching*. Boston: Addison-Wesley, 2002, 983 s. ISBN 0-201-39861-3.

[12] *RFC 1058: RIP protocol* [online]. 1988 [cit. 2019-12-16]. Dostupné z: https://tools.ietf.org/html/rfc1058

[13] *RFC 1247: OSPF protocol* [online]. [cit. 2019-12-16]. Dostupné z: https://tools.ietf.org/html/rfc1247

[14] WONG, Angus a Alan YEUNG. Network Infrastructure Security — Routing. *Network Infrastructure Security*. Boston, MA: Springer US, 2009, s. 59-135. DOI: 10.1007/978-1-4419-0166-8_3. ISBN 9781441901651.

[15] MINOLI, Daniel. *IP multicast with applications to IPTV and mobile DVB-H*. Hoboken: John Wiley, 2008, xvi, 357 s. : il. ISBN 978-0-470-25815-6.

[16] WANG, Feng, Lixin GAO, Charles R KALMANEK, Sudip MISRA a Yang YANG. Interdomain Routing and Reliability. *Guide to Reliable Internet Services and Applications*. London: Springer London, 2010, s. 181-220. DOI: 10.1007/978-1-84882-828-5_6. ISBN 9781848828278.

[17] CISCO NETWORKING ACADEMY PROGRAM. *Scaling networks v6: companion guide*. 2. Indianapolis, IN: Cisco press, 2018, xxiii, ISBN 978-1-58713-434-0.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

GNS3	qGraphical Network Simulator 3
VPCS	Virtual PC Simulator
DHCP	Dynamic Host Configuration Protocol
QoS	Quality of service
IP	Internet Protocol
GUI	Graphical user interface
VM	Virtual machine
Cisco IOS	Cisco Internetwork Operating System
RAM	Random-access memory
CPU	central processing unit
KVM	Kernel-based Virtual Machine
USB	Universal Serial Bus
API	Application programming interface
VT-x	Intel virtualization
PC	Personal computer
OS	Operating system
TFTP	Trivial File Transfer Protocol
SSH	Secure shell
VLAN	Virtual LAN
PPPoE	Point-to-Point Protocol over Ethernet
PPTP	Point-to-Point Tunneling Protocol
L2TP	Layer 2 Tunneling Protocol
VRF	Virtual Routing and Forwarding
ECMP	Equal-cost multi-path routing
FIB	Forwarding information base
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
BGP	Border Gateway Protocol

IPv6	Internet Protocol version 6
IPv4	Internet Protocol version 4
BFD	Bidirectional Forwarding Detection
DNS	Domain Name System
PCQ	Per Connection Queue
HTB	Hierarchical Token Bucket
EIGRP	Enhanced Interior Gateway Routing Protocol
IGP	Interior Gateway Protocol
UDP	User Datagram Protocol
SPF	Dijkstra's Shortest Path First algorithm
AS	Autonomous System
TCP	Transmission Control Protocol
IGMP	Internet Group Management Protocol
PIM	Protocol Independent Multicast
ARP	Address Resolution Protocol
DOS	Denial-of-service attack
SSD	Solid-state drive
ICMP	Internet Control Message Protocol
CHR	Cloud Hosted Router
RTT	Round-trip time